Programmer's Wizard 2 [Nápovìda]

Hledáte odpovìï na konkrétní dotaz? Zkuste hledat zde:
Pøíruèka uživatele

Tip: Kliknutím na logo na horní stranì stránky se vždy dostanete zpìt do obsahu.

Programmer's Wizard 2
http://pwiz.hyperlink.cz/

Licence

Tento program je v souèasné verzi freeware. Autor nenese žádnou odpovìdnost za pøípadné škody, ztráty dat nebo finanèní ztráty zpùsobené používáním tohoto programu. Program smí být dále redistribuován pouze v originálním archivu, který obsahuje funkèní instalátor a soubory README. Redistribuce pomocí Internetu a CD-ROMù je možná jen se svolením autora. Pøi redistribuci nesmí být originální archiv jakkoliv zmìnìn (tj. není možné mìnit, odebírat nebo pøidávat soubory). Program nemùže být dekompilován nebo pøevádìn ze strojového kódu a nesmí být provedeny žádné zmìny v jeho binárních souborech ani v souborech README bez svolení autora. Uživatel mùže provádìt instalace programu na libovolný poèet poèítaèù, pokud k instalaci použije originální archivy, pøièemž registrovat musí jen první kopii programu. Pokud bude program na jednom poèítaèi používat více osob, vyplní a odešle registraèní formuláø každý z uživatelù.

Autor

Autor

Petr Ešner
petr.esner@atlas.cz
#ICQ 62399997

Co je Programmer's Wizard 2?

Programmer's Wiazrd 2 je freewarový nástroj urèený programátorùm, vývojáøùm a pokroèilým uživatelùm. Lze jej definovat jako univerzální editor pro soubory mnoha formátù, bìžící na platformì **Win9x** a **WinNT**. Jedná se o MDI aplikaci s mnoha editory a monitory.

Jádro programu Programmer's Wizard 2 pøedpokládá spolupráci s více vnìjšími zásuvnými moduly (add-ons, plugins...jak kdo chce). Distribuce programu obsahuje tyto komponenty:

- **Program samotný**
  Tato èást je spoleèná pro všechny nástroje, editory a monitory. Stará se o uživatelské rozhraní, spoleèné funkce, obsluhu skriptování apod.
- **Zásuvné moduly**
  Obsahují vlastní implementace jednotlivých nástrojù. Každý zásuvný modul je reprezentován knihovnou (.dll), která mùže obsahovat jeden nebo více nástrojù. Tato verze programu je instalována s následujícími moduly:
- **PWEDIT.DLL** – nástroje SyntaxEditor a HexEditor
- **PWSYSMON.DLL** – Monitor procesù, Monitor vláken, Process Navigator
- **PWSCRIPT.DLL** – Editor skriptù

Tato verze programu (s moduly PWEDIT, PWSYSMON a PWSCRIPT) nabízí:

SyntaxEditor:

Textový editor s barvením syntaxe, v souèasné verzi je podporováno 33 rùzných programovacích a skriptovacích jazykù (C++, Object Pascal, Java, Fortran, Visual Basic, HTML, CSS, PHP, VBScript, JavaScript, AWKScript, Perl, Tcl Tk, Python, CA-Clipper, Foxpro, SQL, Sybase SQL, x86 Assembler a jiné, také je zde podpora .INI konfiguraèních souborù, .BAT dávkových souborù MS-DOSu, formuláøù Delphi a C++ Builder..) SyntaxEditor podporuje nìkolik druhù záložek, standardní operace se schránkou, operace Zpìt/Opakovat, zobrazování èísel øádkù, prohledávání dokumentu (hledat lze øetìzce, 12 možností hledání), funkce 'Najdi a nahraï'. Je zde plná nastavitelnost barvení syntaxe / barevných schémat, editor podporuje Drag&Drop operace. Oznaèený blok lze vložit do jiného souboru, jiný soubor lze vložit do právì upravovaného. Vždy je k dispozici koncextová nabídka s pøíhodnými pøíkazy. Upravovaný dokument lze pøepnout do režimu 'Jen pro ètení', kdy je zamezeno jeho nechtìné zmìnì.

HexEditor:

Hexa editor s vlastnostni textového editoru. Zobrazuje pole ve formátu [Adresa:Data:ASCII data], šíøka zobrazovacích polí a tím i poèet bytù na øádku lze nastavit jednoduchým pøetažením okna. Hexa editor mùže vkládat/odstraòovat znaky, mìnit data v souboru pomocí metody In-place editing. Je zde standardní kurzor, jaký bývá u textových editorù. Hexa editor podporuje operace se chránkou, má neomezené množství krokù u funkcí Zpìt/Opakovat. Obsahuje funkce pro hledání dat; hledat lze ASCII øetìzce, binární hodnoty i promìnné, podporováno je 9 standardních datových typù. Prohledávání rozsáhlých dokumentù mùže vygenerovat i desítky tisíc výsledkù s minimální závislostí rychlosti programu na poètu zpracovávaných výsledkù. Výsledky hledání mùžou zùstat zobrazeny v pomocném panelu, odkud lze pøecházet na jejich místa v dokumentu nebo je upravovat. Dokument lze pøepnout do režimu 'Jen pro ètení', kdy je zamezeno jeho nechtìné zmìnì.

Editor skriptù:

Editor skriptù je urèen pro úpravu skriptù programu Programmer's Wizard 2. Pomocí skriptù si mùžete zcela pøizpùsobit chování programu v urèitých situacích. Mùžete si napsat vlastní skript, který za Vás bude provádìt èasté operace nebo který se bude starat o Vaše dokumenty. Editor skriptù je textovým editorem s barvením syntaxe zamìøeným na tento skriptovací jazyk. Detaily o jazyku naleznete po instalaci v souboru SCRIPT.DOC nebo v této nápovìdì v èásti Skriptování. Další funkce Editoru skriptù jsou zcela identické s funkcemi SyntaxEditoru, vše, co lze v SyntaxEditoru, je možné i tady. Editor skriptù navíc obsahuje funkce AutoComplete a CompletionProposal, které Vám pomohou s psaním Vašich

vlastních skriptù.

Monitor procesù:

Monitor procesù je nástroj urèený pro sledování bìžících procesù. Vypisuje jejich seznam, technické parametry a další informace. Výsledky sledování lze uložit do souboru nebo zkopírovat do schránky.

Monitor vláken:

Monitor vláken je dalším nástrojem, který má za úkol sledovat systémové pochody. Tento monitor vypisuje seznam existujících vláken (spuštìných i ukonèených), jejich technické parametry a další informace. Výsledky sledování lze také uložit do souboru, popøípadì zkopírovat do schránky.

Process Navigator (nyní dostupné i na WinNT 5.0 pro uživatele skupiny Guests!) **:**

Tento nástroj mùže sledovat libovolný bìžící proces. Zobrazí jeho vlákna a jejich stav (priorita, ID, handle); knihovny a moduly, které má program naètené, soubor, ze kterého byly naèteny a další detaily. Tento monitor navíc umí pracovat s pamìtí sledovaného procesu. Mùžete prohledávat pamì na øetìzce, binární data nebo promìnné, nalezená data pak mùžete upravovat pøímo v pamìti sledovaného procesu za jeho bìhu (!). **Upozornìní:** tento nástroj mùže být (jako jediný) skuteènì nebezpeèný, pokud nevíte, co dìláte. Sledovaný proces nemùže zjistit, že byla nìjak narušena jeho pamìt a že jeho soukromá data byla zmìnìna. Neuvážený zápis do pamìti procesù mùže zpùsobit nekonzistentnost struktur ve sledovaném procesu a tím i jeho nestabilitu, pøípadnì nestabilitu celého systému!

Programmer's Wizard 2 používá uživatelské rozhraní standardu Microsoft Office; setkáte se tedy s panely nástrojù, animovanými grafickými nabídkami i dalšími prvky. Uživatelskému konfortu pøispívá i vysoká nastavitelnost témìø všech vlastností programu. Mnoho dalších funkcí (jako jsou 'Automatické zálohování', 'Podpora externích programù', 'Podpora projektù'…) pøispívají k plnému pohodlí a zvyšují efektivitu práce v programu.

Zásuvné moduly

Jádro programu Programmer's Wizard 2 pøedpokládá spolupráci s více vnìjšími zásuvnými moduly (add-ons, plugins...jak kdo chce). Distribuce programu obsahuje tyto zásuvné moduly (dále jen *moduly*):

- pwedit.dll
- pwsysmon.dll
- pwscript.dll

Každý z *modulù* obsahuje jeden nebo více nástrojù. Programmer's Wizard 2 po spuštìní prohledá pøíslušný adresáø s *moduly* a najde všechny knihovny, které by mohly být jeho *moduly*. Pak všechny nalezené moduly inicializuje a pøipraví k práci. Naètení modulù mùže zpùsobit zmìnu nabídek programu (pøidání pøíkazù) nebo zmìnu panelù nástrojù (pøidání tlaèítek). Po úspìšném naètení všech modulù je program pøipraven k práci.
Tato architektura programu má oproti staticky vytvoøenému programu nìkolik výhod:

- když bude k dispozici nová verze nìkterého nástroje, nemusíte si stahovat program celý; postaèí stáhnout jen aktualizovaný modul
- program je otevøený k novým modulùm, když bude k dispozici nový nástroj, staèí stáhnout modul s nástrojem a program jej zaène používat
- nemusíte mít naètené všechny moduly najednou, pokud nepoužíváte nìkterý nástroj, mùžete odstranit modul s tímto nástrojem a tím urychlit dobu naèítání programu

Win9x

Windows 95, Window 98, Window98 SE

WinNT

WinNT 4.0, WinNT 5.0 (Windows 2000)

WinNT 5.0

Windows 2000

Knihovny

Binární soubory v spustitelném formátu, které obsahují kód a data. Jsou to souèásti jiných programù, které je používají ke své práci. Název souboru mívá obvykle koncovku .dll (ale není to nutností).

.dll

Zkratka od *Dynamic Link Library* (pø. dynamicky linkovaná knihovna)

Technické požadavky

Programmer's Wizard 2 lze spustit na platformì Win9x a WinNT. Program nevyžaduje žádný nestandardní hadrware a pro plné využití funkcí postaèí tato konfigurace:

Pentium 90
32MB RAM
Windows 95
Zobrazení min. 800x600 pøi 256 barvách

Technické specifikace

Program Programmer's Wizard 2 a všechny jeho zásuvné moduly jsou 32-bitové aplikace vytvoøené pomocí vývojového prostøedí Borland (Inprise) Delphi 5. Program nepoužívá žádné speciální verze knihoven Windows, které nejsou k dispozici na Windows95.
Program nelze spustit na Win3x s balíkem Win32s, který integruje èásteènou 32bitovou podporu do Windows verzí 3x.

Technické detaily

Odkazy:
Technické požadavky
Technické specifikace

pwedit.dll
Zásuvný modul

Obsahuje nástroje:
SyntaxEditor
HexEditor

pwsysmon.dll
Zásuvný modul

Obsahuje nástroje:
- Monitor procesù
- Monitor vláken
- Process Navigator

pwscript.dll
Zásuvný modul

Obsahuje nástroje:
 Editor skriptù

SyntaxEditor (struènè shrnutí)
(modul pwedit.dll)

Textový editor s barvením syntaxe, v souèasné verzi je podporováno 33 rùzných programovacích a skriptovacích jazykù (C++, Object Pascal, Java, Fortran, Visual Basic, HTML, CSS, PHP, VBScript, JavaScript, AWKScript, Perl, Tcl Tk, Python, CA-Clipper, Foxpro, SQL, Sybase SQL, x86 Assembler a jiné, také je zde podpora .INI konfiguraèních souborù, .BAT dávkových souborù MS-DOSu, formuláøù Delphi a C++ Builder..) SyntaxEditor podporuje nìkolik druhù záložek, standardní operace se schránkou, operace Zpìt/Opakovat, zobrazování èísel øádkù, prohledávání dokumentu (hledat lze øetìzce, 12 možností hledání), funkce 'Najdi a nahraï'. Je zde plná nastavitelnost barvení syntaxe / barevných schémat, editor podporuje Drag&Drop operace. Oznaèený blok lze vložit do jiného souboru, jiný soubor lze vložit do právì upravovaného. Vždy je k dispozici koncextová nabídka s pøíhodnými pøíkazy. Upravovaný dokument lze pøepnout do režimu 'Jen pro ètení', kdy je zamezeno jeho nechtìné zmìnì.

Odkazy:
Podporované programovací a skriptovací jazyky

HexEditor (struèné shrnutí)
(modul pwedit.dll)

Hexa editor s vlastnostni textového editoru. Zobrazuje pole ve formátu [Adresa:Data:ASCII data], šíøka zobrazovacích polí a tím i poèet bytù na øádku lze nastavit jednoduchým pøetažením okna. Hexa editor mùže vkládat/odstraòovat znaky, mìnit data v souboru pomocí metody In-place editing. Je zde standardní kurzor, jaký bývá u textových editorù. Hexa editor podporuje operace se chránkou, má neomezené množství krokù u funkcí Zpìt/Opakovat. Obsahuje funkce pro hledání dat; hledat lze ASCII øetìzce, binární hodnoty i promìnné, podporováno je 9 standardních datových typù. Prohledávání rozsáhlých dokumentù mùže vygenerovat i desítky tisíc výsledkù s minimální závislostí rychlosti programu na poètu zpracovávaných výsledkù. Výsledky hledání mùžou zùstat zobrazeny v pomocném panelu, odkud lze pøecházet na jejich místa v dokumentu nebo je upravovat. Dokument lze pøepnout do režimu 'Jen pro ètení', kdy je zamezeno jeho nechtìné zmìnì.

Monitor procesù (struènè shrnutí)
(modul pwsysmon.dll)

Monitor procesù je nástroj urèený pro sledování bìžících procesù. Vypisuje jejich seznam, technické parametry a další informace. Výsledky sledování lze uložit do souboru nebo zkopírovat do schránky.

Monitor vláken (struèné shrnutí)
(modul pwsysmon.dll)

Monitor vláken je dalším nástrojem, který má za úkol sledovat systémové pochody. Tento monitor vypisuje seznam existujících vláken (spuštìných i ukonèených), jejich technické parametry a další informace. Výsledky sledování lze také uložit do souboru, popøípadì zkopírovat do schránky.

Process Navigator (struèné shrnutí)
(modul pwsysmon.dll)

Tento nástroj mùže sledovat libovolný bìžící proces. Zobrazí jeho vlákna a jejich stav (priorita, ID, handle); knihovny a moduly, které má program naètené, soubor, ze kterého byly naèteny a další detaily. Tento monitor navíc umí pracovat s pamìtí sledovaného procesu. Mùžete prohledávat pamì na øetìzce, binární data nebo promìnné, nalezená data pak mùžete upravovat pøímo v pamìti sledovaného procesu za jeho bìhu (!). **Upozornìní:** tento nástroj mùže být (jako jediný) skuteènì nebezpeèný, pokud nevíte, co dìláte. Sledovaný proces nemùže zjistit, že byla nìjak narušena jeho pamìt a že jeho soukromá data byla zmìnìna. Neuvážený zápis do pamìti procesù mùže zpùsobit nekonzistentnost struktur ve sledovaném procesu a tím i jeho nestabilitu, pøípadnì nestabilitu celého systému!

Editor skriptù (struèné shrnutí)
(modul pwscript.dll)

Editor skriptù je urèen pro úpravu skriptù programu Programmer's Wizard 2. Pomocí skriptù si mùžete zcela pøizpùsobit chování programu v urèitých situacích. Mùžete si napsat vlastní skript, který za Vás bude provádìt èasté operace nebo který se bude starat o Vaše dokumenty. Editor skriptù je textovým editorem s barvením syntaxe zamìøeným na tento skriptovací jazyk. Detaily o jazyku naleznete po instalaci v souboru SCRIPT.DOC nebo v této nápovìdì v èásti Skriptování. Další funkce Editoru skriptù jsou zcela identické s funkcemi SyntaxEditoru, vše, co lze v SyntaxEditoru, je možné i tady. Editor skriptù navíc obsahuje funkce AutoComplete a CompletionProposal, které Vám pomohou s psaním Vašich vlastních skriptù.

Nástroje

Odkazy:
- pwedit.dll
- SyntaxEditor
- HexEditor
- pwsysmon.dll
- Monitor procesù
- Monitor vláken
- Process Navigator
- pwscript.dll
- Editor skriptù

Podporované programovací jazyky

SyntaxEditor podporuje tyto programovací a skriptovací jazyky:

- 68HC11
- ADSP21xx
- AWK skript
- Baan 4GL
- C++
- CA-Clipper
- CSS (Cascading Stylesheets)
- Delphi / C++ Builder formuláøe
- Fortran
- Foxpro
- Galaxy
- Gembase
- HP48
- HTML
- INI konfiguraèní soubory
- Java
- JavaScript
- KIX32
- MS VBScript
- MS-DOS dávkový soubor
- ObjectPascal
- Perl
- PHP
- Python
- Resource files (soubory zdrojù .rc)
- SQL
- Standard ML
- Sybase SQL
- Tcl Tk
- Visual Basic
- x86 Assembly Language

Nastavení adresáøù programu

Programmer's Wizard 2 pracuje s tìmito výchozími adresáøi:

složka s projekty
Zde se ukládají projekty, vytvoøené programem. Výchozí umístìní: \Projects

složka se zásuvnými moduly
V této složce jsou zásuvné moduly používané programem. Výchozí umístìní: \Plugins

složka se skripty
Tady jsou skripty, vytvoøené programem. Výchozí umístìní: \Scripts

Umístìní tìchto složek lze zmìnit z dialogu *Souštìní a ukonèování*, který lze otevøít vybráním stejnojmeného pøíkazu z hlavní nabídky 'Soubor' / 'Nastavení'. Zmìny se projeví pøi pøíštím spuštìní programu.

Odkazy:
Struktura složky programu

Struktura složky programu

Toto je adresáøová struktura programu tak, jak se vytvoøí po instalaci programu. Kliknutím na jednotlivé ikony získáte jejich popis.

Programmer's Wizard 2
- **Plugins**
  - **pwedit.dll**
  - **pwscript.dll**
  - **pwsysmon.dll**
- **Projects**
- **Scripts**
- **borlndmm.dll**
- **pwrel32.exe**
- **readme.doc**
- **readme.txt**

Odkazy:
Umístìní složek **Plugins**, **Projects** a **Scripts** lze kdykoliv zmìnit.

readme.txt

Základní informace o programu Programmer's Wizard 2, shrnutí funkcí, licence a další informace. Vše v èistì textovém formátu.

readme.doc

Základní informace o programu Programmer's Wizard 2, shrnutí funkcí, licence a další informace. Urèeno pro MS Word 97 a vyšší.

pwrel32.exe

Vlastní program Programmer's Wizard 2.

borlndmm.dll

Knihovna 'Správce sdílené pamìti'

Projekty

Programmer's Wizard 2 obsahuje podporu projektù, pomocí kterých si mùžete pøehlednì uspoøádat práci.

Projekt je seznam souborù, které patøí k urèité skupinì. Napøíklad, projekt s WWW prezentací bude obsahovat všechny soubory, které jsou k prezentaci nutné - tedy všechny její stránky, skripty i grafiku. Všechny tyto soubory jsou nutné pro prezentaci. Analogicky, dejme je všechny do jedné *krabice* s popisem "Projekt WWW prezentace", protože pak nebude potøeba prohledávat spousty dokumentù na stole, staèí jít rovnou do krabice. Mùžeme si takto rozdìlit všechnu práci a popsané krabice dát .. tøeba na polici.

V programu je pak touto policí nabídka 'Projekt'. Z této nabídky lze otevírat projekty, vytváøet nové nebo upravovat existující.

Pøi vytváøení projektu se zobrazí dialog 'Upravit projekt', ve kterém je potøeba pøidat do projektu soubory, které má obsahovat. Po vybrání souborù, zadání jména a popisu projektu se projekt automaticky otevøe a také se zavøou všechny okna se soubory, které byly otevøeny pøed vytvoøením projektu. Mùžete zaèít pracovat. K otevøení souboru z projektu staèí vybrat pøíkaz 'Otevøít dokument' z nabídky 'Projekt' - zobrazí se seznam všech souborù, které jsou v projektu. Jednoduchým kliknutím na jméno souboru je pøíslušný soubor otevøe a pøipraví k editaci.

Až bude potøeba projekt zavøít, nezavírejte napøed všechny otevøené dokumenty, místo toho vyberte pøíkaz 'Zavøít projekt' z nabídky 'Projekt'. Okna se zavøou všechna najednou, pokud je nìkterý soubor zmìnìn, zobrazí program upozornìní a mùžete zmìny uložit. Hlavní výhoda používání projektù spoèívá v tom, že program pøi zavírání projektu uloží také jeho stav (otevøené soubory, polohu jejich oken atd.) a pøi pøíštím otevøení projektu je obnoví a tak lze pokraèovat v práci od místa, kde se pøestalo. Další výhodou použití projektù je snadný pøístup k souborùm; není nutné otevírat okna a v nich hledat potøebný soubor, staèí se proklikat jednoduchou nabídkou.

Odkazy:
Plochy

Plochy

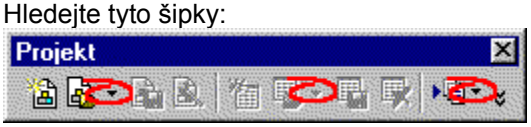
Plochy jsou souèástí projektù. Každý projekt mùže obsahovat nìkolik ploch. Pøi vytvoøení projektu je také vytvoøena jeho primární plocha.

V projektu programu Programmer's Wizard 2 je plochou nazývána ta èást okna, ve které se zobrazují upravované dokumenty a okna nástrojù. *Stavem plochy* je rozumìn seznam otevøených souborù, poloha jejich oken, stav nástrojù použitých k jejich editaci apod. Programmer's Wizard 2 pøi zavírání projektu zavøe jeho primární plochu a uloží její *stav*. Pøi pøíštím otevøením projektu pak otevøe primární plochu a stav obnoví, tj. otevøe opìt všechny dokumenty, které byly otevøeny pøed zavøením plochy a obnoví nástroje do takového stavu, v jakém pøed zavøením byly.

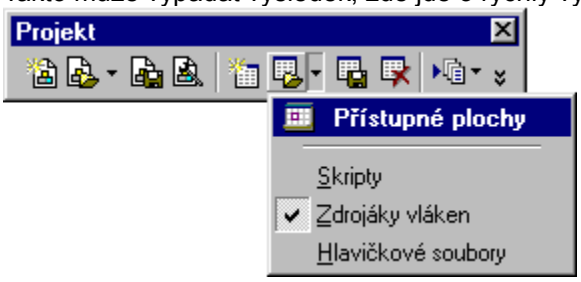
Projekt programu Programmer's Wizard 2 mùže však obsahovat více ploch najednou, každou se svým vlastním *stavem*. Pokud budeme vycházet z pøíkladu WWW prezentace, probraného v tématu Projekty, mùžeme primární plochu nazvat 'HTML' a otevøít do ní nìkolik HTML stránek. Pak lze vytvoøit novou plochu s názvem 'Skripty', ve které budeme pracovat jen se skripty. Hlavní výhodou takového rozdìlení práce je to, že *každá* plocha bude pøi uzavøení uložena. Když tedy budete chtít pracovat se skripty, otevøete stejnojmenou plochu a upravíte skript. Pak se lze pøepnout zpìt na plochu 'HTML' a pokraèovat v úpravì stránek tam, kde se pøestalo.

Projekty a plochy programu Programmer's Wizard 2 zpøehledòují práci. Pøi jejich použití mùže být zdlouhavé používání dialogu 'Otevøít' minulostí.

Pozn.: pro nejvyšší pohotovost pøístupu k plochám a funkcím projektù používejte místo nabídky radìji panel nástrojù 'Projekty'. Tento zpùsob je zvláštì vhodný pro rychlé otevírání ploch a souborù, protože vedle nìkterých tlaèítek na panelu nástrojù se zobrazuje šipka, po jejímž stisknutí se zobrazí pøíhodná nabídka.
Hledejte tyto šipky:



Takto mùže vypadat výsledek, zde jde o rychlý výbìr plochy:



Odkazy:
Projekty

Panely nástrojù

Programmer's Wizard 2 obsahuje nìkolik panelù nástrojù, které zrychlují a zpøíjemòují práci s programem. Obsahují nejbìžnìjší pøíkazy z nabídek a jsou uživatelsky konfigurovatelné (umístìní, poèet tlaèítek, jejich význam). Nìkteré funkce pro zrychlení práce jsou dostupné dokonce jen z panelu nástrojù (viz. šipky v tématu Plochy).

Vyberte panel, o kterém se chcete dozvìdìt více:

- Nabídky
- Soubor
- Úpravy
- Poznámky
- Okno
- Formát
- Projekt
- Záložky
- Dokumenty
- Skripty

- Poznámky (panel)
- Složka Dokumenty (panel)

Odkazy:
- Vlastní panely nástrojù

Panel 'Soubor'

Panel nástrojù 'Soubor' dává pøístup k dùležitým pøíkazùm ze stejmojmené nabídky.
Zde je obrázek panelu nástrojù. Kliknutím na jeho èásti se dozvíte více.

(kliknout zde)

Odkazy:
Panely nástrojù

Soubor | Nový

Tento pøíkaz vytvoøí nový soubor použitím výchozího editoru. Pokud není nainstalován žádný výchozí editor, zobrazí se okno se seznamem všech možných typù dokumentù.

Soubor | Otevøít

Tento pøíkaz otevøe dialogové okno 'Otevøít', ve kterém lze vybrat soubor, který se má otevøít. Vybraný soubor se pak otevøe v pøíslušném nástroji.

Soubor | Nový (šipka)

Tato šipka zobrazí po svém stisknutí nabídku se seznamem všech typù souborù, které mohou být vytvoøeny. Obsah nabídky záleží na nainstalovaných modulech.

Soubor | Uložit

Po vybrání tohoto pøíkazu se uloží právì upravovaný dokument.

Soubor | Uložit vše

Po vybrání tohoto pøíkazu se uloží všechny právì upravované zmìnìné soubory.

Soubor | Tisk

Po vybrání tohoto pøíkazu se vytiskne právì upravovaný doukment, pokud to daný nástroj umožòuje.

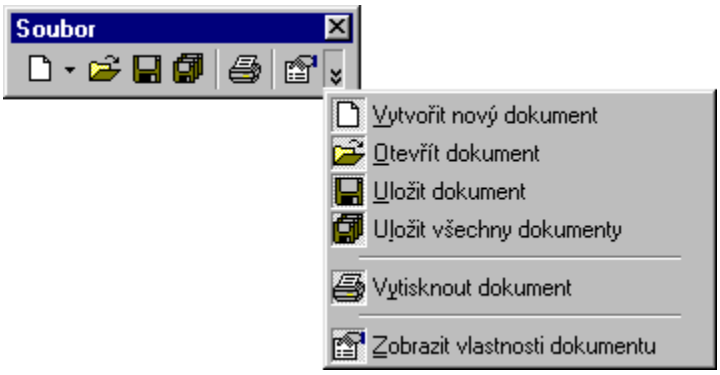
Soubor | Vlastnosti

Po vybrání tohoto souboru se zobrazí vlastnosti právì upravovaného souboru (jméno, velikost, atributy). Ze zobrazeného okna lze také vlastnosti souboru mìnit.

Vlastní panely nástrojù

Panely nástrojù programu Programmer's Wizard 2 mohou být pøizpùsobeny každému (témìø každému :-)) uživateli. K nastavení panelù lze použít dvì techniky:

Volba tlaèítek panelu vybráním tlaèítka 'Šipka' (, viz napø. panel Soubor), které je na konci každého panelu.
Po stisku šipky se zobrazí nabídka se seznamem všech pøíkazù, které jsou na panelu k dispozici.



Odznaèením jednotlivých pøíkazù se pøíslušná tlaèítka ukryjí, jejich opìtným oznaèením se tlaèítka zviditelní.

Dialog 'Upravit panely nástrojù'
Tento dialog nabízí nastavení všech vlastností všech panelù z jednoho místa. Dialog 'Upravit panely nástrojù' lze otevøít vybráním stejnojmeného pøíkazu z nabídky, které se otevøe po kliknutí pravým myšítkem na kterýkoliv panel.

Panel 'Úpravy'

Panel nástrojù 'Úpravy' dává pøístup k dùležitým pøíkazùm ze stejmojmené nabídky.
Zde je obrázek panelu nástrojù. Kliknutím na jeho èásti se dozvíte více.

(kliknout zde)

Odkazy:
Panely nástrojù

Úpravy | Vyjmout

Vyjme vybraná data do schránky.

Úpravy | Kopírovat

Zkopíruje vybraná data do schránky.

Úpravy | Vložit

Vloží do dokumentu data ze schránky.

Úpravy | Zpìt

Vrátí poslední provedenou operaci.

Úpravy | Zpìt (šipka)

Zobrazí seznam provedených operací, které lze vrátit, pokud to pøislušný nástroj dovoluje.

Úpravy | Opakovat

Zopakuje poslední vrácenou operaci.

Úpravy | Opakovat (šipka)

Zobrazí seznam vrácených operací, které lze zopakovat, pokud to pøislušný nástroj dovoluje.

Úpravy | Najít

Zobrazí okno, pomocí kterého je možné zaèít prohledávání dokumentu.

Úpravy | Nahradit

Zobrazí okno, pomocí kterého je možné nahradit urèitá data v dokumentu.

Panel 'Nabídky'

Panel nástrojù 'Nabídky' dává pøístup ke všem pod-nabídkám programu.
Zde je obrázek panelu nástrojù. Kliknutím na jeho èásti se dozvíte více.

(kliknout zde)

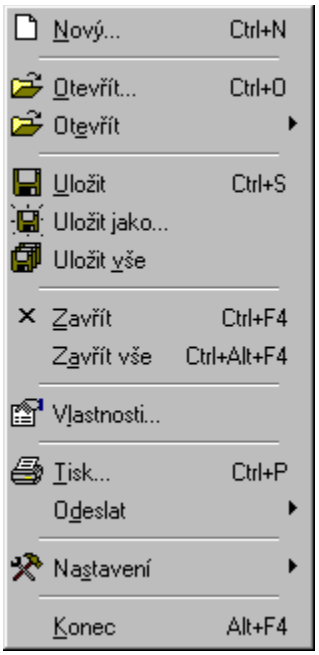
..nebo mùžete kliknout na tyto názvy jednotlivých nabídek:

Soubor
Úpravy
Zobrazit
Formát
Projekt
Dokumenty
Spustit
Skript
Nástroje
Okno
Nápovìda

Odkazy:
Panely nástrojù

Nabídka 'Soubor'

Nabídka 'Soubor' se otevøe po kliknutí na stejnojmený pøíkaz na panelu nabídek.
Zde je obrázek nabídky. Kliknutím na jednotlivé èásti se dozvíte více.

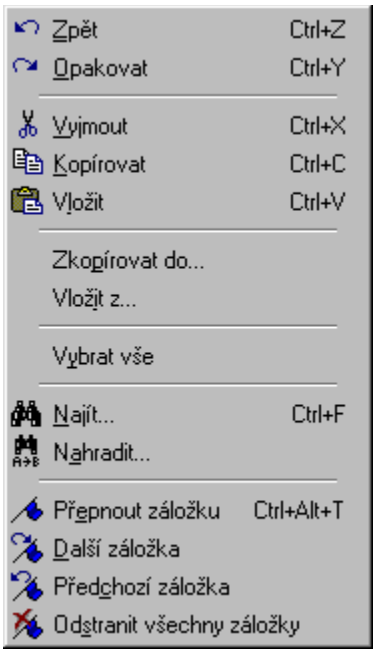


(kliknout zde)

Odkazy:
Panely nástrojù

Nabídka 'Úpravy'

Nabídka 'Úpravy' se otevøe po kliknutí na stejnojmený pøíkaz na panelu nabídek.
Zde je obrázek nabídky. Kliknutím na jednotlivé èásti se dozvíte více.

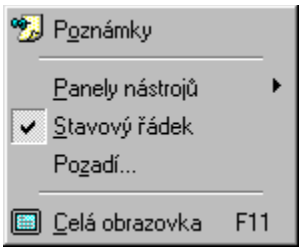


(kliknout zde)

Odkazy:
Panely nástrojù

Nabídka 'Zobrazit'

Nabídka 'Zobrazit' se otevøe po kliknutí na stejnojmený pøíkaz na panelu nabídek.
Zde je obrázek nabídky. Kliknutím na jednotlivé èásti se dozvíte více.

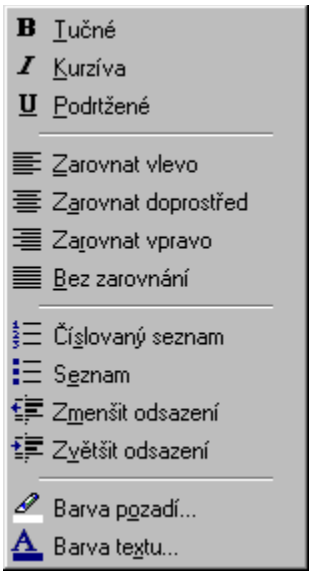


(kliknout zde)

Odkazy:
Panely nástrojù

Nabídka 'Formát'

Nabídka 'Formát' se otevøe po kliknutí na stejnojmený pøíkaz na panelu nabídek.
Zde je obrázek nabídky. Kliknutím na jednotlivé èásti se dozvíte více.

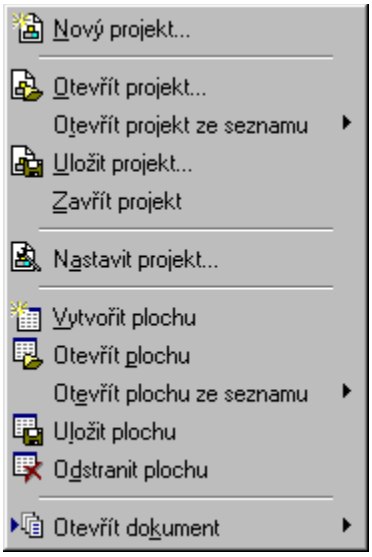


(kliknout zde)

Odkazy:
Panely nástrojù

Nabídka 'Projekt'

Nabídka 'Projekt' se otevøe po kliknutí na stejnojmený pøíkaz na panelu nabídek.
Zde je obrázek nabídky. Kliknutím na jednotlivé èásti se dozvíte více.

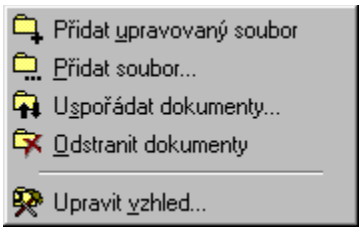


(kliknout zde)

Odkazy:
Panely nástrojù

Nabídka 'Dokumenty'

Nabídka 'Dokumenty' se otevøe po kliknutí na stejnojmený pøíkaz na panelu nabídek.
Zde je obrázek nabídky. Kliknutím na jednotlivé èásti se dozvíte více.



(kliknout zde)

Odkazy:
Panely nástrojù

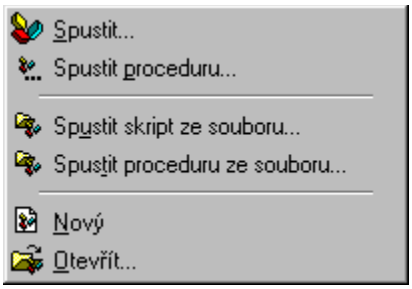
Nabídka 'Spustit'

Tato nabídka je portálem nainstalovaných zásuvných modulù. Obsahuje všechny nástroje, které lze spustit. Kliknutím na libovoulný pøíkaz v této nabídce se otevøe pøíslušný nástroj.

Odkazy:
Panely nástrojù

Nabídka 'Skript'

Nabídka 'Skript' se otevøe po kliknutí na stejnojmený pøíkaz na panelu nabídek.
Zde je obrázek nabídky. Kliknutím na jednotlivé èásti se dozvíte více.

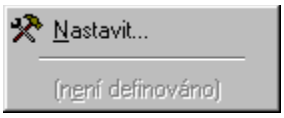


(kliknout zde)

Odkazy:
Panely nástrojù

Nabídka 'Nástroje'

Nabídka 'Nástroje' se otevøe po kliknutí na stejnojmený pøíkaz na panelu nabídek.
Zde je obrázek nabídky. Kliknutím na jednotlivé èásti se dozvíte více.



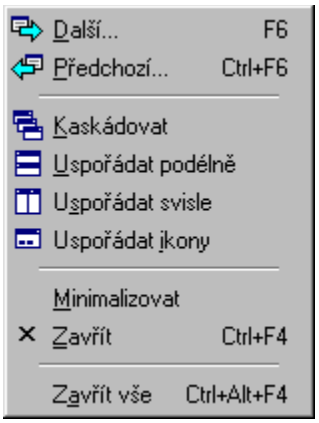
(kliknout zde)

Tato nabídka je uživatelsky definovaná, viz. Externí nástroje.

Odkazy:
Externí nástroje
Panely nástrojù

Nabídka 'Okno'

Nabídka 'Okno' se otevøe po kliknutí na stejnojmený pøíkaz na panelu nabídek.
Zde je obrázek nabídky. Kliknutím na jednotlivé èásti se dozvíte více.

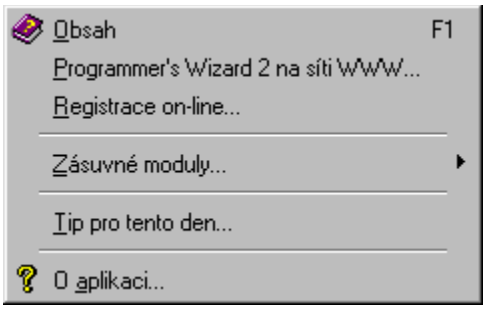


(kliknout zde)

Odkazy:
Panely nástrojù

Nabídka 'Nápovìda'

Nabídka 'Nápovìda' se otevøe po kliknutí na stejnojmený pøíkaz na panelu nabídek.
Zde je obrázek nabídky. Kliknutím na jednotlivé èásti se dozvíte více.



(kliknout zde)

Odkazy:
Panely nástrojù

Soubor | Otevøít (nedávno otevøené soubory)

Zobrazí nabídku se seznamem naposledy použitých dokumentù. Kliknutím na nìkterý dokument se tento otevøe v pøíslušném nástroji.

Soubor | Uložit jako

Uloží právì upravovaný dokument pod novým, vybraným názvem.

Soubor | Zavøít

Umožní uložení zmìn v právì otevøeném dokumentu pak jej uzavøe.

Soubor | Zavøít vše

Umožní uložení zmìn ve všech právì otevøených dokumentech pak je uzavøe.

Soubor | Odeslat

Zobrazí podnabídku se seznamem míst, na která lze odeslat právì otevøený dokument.

Soubor | Nastavení

Otevøe podnabídku se seznamem funkcí, které lze nastavit.

Soubor | Konec

Uzavøe projekt (pokud je otevøen), všechna okna a pak ukonèí program.

Úpravy | Vybrat vše

Oznaèí všechna data v dokumentu.

Úpravy | Zkopírovat do

Zkopíruje vybraná data do jiného dokumentu.

Úpravy | Vložit z

Vloží do dokumentu data z jiného souboru.

Úpravy | Pøepnout záložku

Pøepne záložku na aktuální pozici.

Úpravy | Další záložka

Pøejde na další nastavenou záložku.

Úpravy | Pøedchozí záložka

Pøejde na pøedchozí nastavenou záložku.

Úpravy | Odstranit všechny záložky

Odstraní všechny záložky v dokumentu.

Zobrazit | Poznámky

Zobrazí nebo ukryje panel Poznámky.

Zobrazit | Panely nástrojù

Zobrazí nabídku se seznamem všech panelù nástrojù, pomocí které lze zapnout nebo vypnout kterýkoliv panel nástrojù. Tato nabídka se také zobrazí po kliknutí pravým myšítkem na kterýkoliv panel nástrojù.

Zobrazit | Stavový øádek

Zobrazí nebo ukryje stavový øádek.

Zobrazit | Pozadí

Zobrazí dialog, ve kterém lze nastavit barvu nebo tapetu pozadí plochy programu.

Zobrazit | Celá obrazovka

Pøepíná mezi zobrazováním pøes celou obrazovku a normálním režimem.

Formát | Tuèné

Nastaví tuèné písmo.

Formát | Kurzíva

Nastaví písmo stylu kurzíva.

Formát | Podtržené

Nastaví podtržené písmo.

Formát | Zarovnat vlevo

Zarovná blok textu nebo dat vlevo.

Formát | Zarovnat doprostøed

Zarovná blok textu nebo dat doprostøed.

Formát | Zarovnat vpravo

Zarovná blok textu nebo dat vpravo.

Formát | Bez zarovnání

Zruší zarovnání bloku textu nebo dat.

Formát | Èíslovaný seznam

Vytvoøí èíslovaný seznam.

Formát | Seznam

Vytvoøí obyèejný seznam.

Formát | Zmenšit odsazení

Zmenší odsazení aktuálních nebo vybraných dat.

Formát | Zvìtšit odsazení

Zvìtší odsazení aktuálních nebo vybraných dat.

Formát | Barva pozadí

Umožní nastavit jinou barvu pozadí.

Formát | Barva textu

Umožní nastavit jinou barvu textu.

Projekt | Nový projekt

Otevøe okno, ve kterém lze vytvoøit a nastavit projekt. Vytvoøený projekt pak otevøe.

Projekt | Otevøít projekt

Zobrazí dialog, ve kterém lze vybrat existující projekt. Vybraný projekt otevøe.

Projekt | Otevøít projekt ze seznamu

Zobrazí nabídku se seznamem všech existujících projektù. Po kliknutí na nìkterý z názvù se otevøe pøíslušný projekt.

Projekt | Uložit projekt

Uloží právì otevøený projekt.

Projekt | Zavøít projekt

Uzavøe právì otevøený projekt.

Projekt | Nastavit projekt

Zobrazí okno umožòující nastavit vlastnosti právì otevøeného projektu.

Projekt | Vytvoøit plochu

Vytvoøí novou plochu právì otevøeného projektu.

Projekt | Otevøít plochu

Umožní otevøít nìkterou z ploch právì otevøeného projektu.

Projekt | Otevøít plochu ze seznamu

Zobrazí nabídku se seznamem všech existujících ploch právì otevøeného projektu. Po kliknutí na nìkterou z položek pak pøíslušnou plochu otevøe.

Projekt | Uložit plochu

Uloží právì otevøenou plochu. Plocha je také automaticky uložena pøi jejím zavírání, tj. i pøi zavírání projektu.

Projekt | Odstranit plochu

Odstraní právì otevøenou plochu.

Projekt | Otevøít dokument

Zobrazí seznam všech souborù, které jsou v právì otevøeném projektu. Po kliknutí na nìkterý z názvù se pøíslušný dokument otevøe.

Dokumenty | Pøidat upravovaný soubor

Pøidá právì upravovaný dokument do složky dokumentù.

Dokumenty | Pøidat soubor

Zobrazí dialog, ve kterém je možné vybrat jeden nebo více dokumentù, které se mají do složky dokumentù pøidat.

Dokumenty | Uspoøádat dokumenty

Umožní uspoøádat obsah složky dokumentù.

Dokumenty | Odstranit dokumenty

Odstraní všechny zástupce ze složky dokumentù.

Dokumenty | Upravit vzhled

Zobrazí okno umožòující zmìnit vzhled složky dokumentù.

Skript | Spustit

Zobrazí okno se seznamem nalezených skriptù, vybraný skript pak spustí.

Skript | Spustit proceduru

Zobrazí okno se seznamem nalezených skriptù, po vybrání skriptu zobrazí dotaz na název procedury, která se má provést. Vybranou proceduru pak spustí.

Skript | Spustit skript ze souboru

Zobrazí dialog, ve kterém lze najít a vybrat skript. Vybraný skript pak spustí.

Skript | Spustit proceduru ze souboru

Zobrazí dialog, ve kterém lze najít a vybrat skript, po vybrání skriptu zobrazí dotaz na název procedury, která se má provést. Vybranou proceduru pak spustí.

Skript | Nový

Vytvoøí nový skript nainstalovaným nástrojem pro práci se skripty.

Skript | Otevøít

Zobrazí dialog, ve kterém lze najít a vybrat skript, vybraný skript pak otevøe v nainstalovaném nástroji pro práci se skripty.

Nástroje | Nastavit

Zobrazí dialog umožòující nastavení externích nástrojù.

Okno | Další

Pøenese do popøedí další okno.

Okno | Pøedchozí

Pøenese do popøedí pøedchozí okno.

Okno | Kaskádovat

Kaskáduje všechna otevøená okna.

Okno | Uspoøádat podélnì

Uspoøádá všechna okna podélnì.

Okno | Uspoøádat svisle

Uspoøádá všechna okna svisle.

Okno | Uspoøádat ikony

Uspoøádá všechna minimalizovaná okna.

Okno | Minimalizovat

Minimalizuje právì otevøené okno.

Okno | Zavøít

Zavøe právì otevøené okno.

Okno | Zavøít vše

Zavøe všechna otevøená okna.

Nápovìda | Obsah

Zobrazí seznam nápovìdy.

Nápovìda | Programmer's Wizard 2 na síti WWW

Pøejde na WWW stránky programu Programmer's Wizard 2.

Nápovìda | Registrace on-line...

Zobrazí registraèní formuláø programu Programmer's Wizard 2.

Nápovìda | Zásuvné moduly

Zobrazí informace o nainstalovaných zásuvných modulech.

Nápovìda | Tip pro tento den

Zobrazí okno Tip dne.

Nápovìda | O aplikaci

Zobrazí informace o aplikaci (verze, licence)

Panel 'Formát'

Panel nástrojù 'Formát' dává pøístup k dùležitým pøíkazùm ze stejmojmené nabídky. Zde je obrázek panelu nástrojù. Kliknutím na jeho èásti se dozvíte více.

(kliknout zde)

Odkazy:

Panely nástrojù

Formát | Písmo

Umožní vybrat písmo.

Formát | Velikost písma

Umožní vybrat velikost písma.

Panel 'Poznámky'

Panel nástrojù 'Poznámky' dává pøístup k dùležitým pøíkazùm z nabídky 'Zobrazit'.
Zde je obrázek panelu nástrojù. Kliknutím na jeho èásti se dozvíte více.

(kliknout zde)

Odkazy:
Panely nástrojù

Zobrazit | Poznámky (šipka)

Zobrazí nabídku s pøíhodnými pøíkazy pro otevøení nebo vymazání poznámek.

Panel 'Záložky'

Panel nástrojù 'Záložky' dává pøístup k dùležitým pøíkazùm z nabídky 'Úpravy'.
Zde je obrázek panelu nástrojù. Kliknutím na jeho èásti se dozvíte více.

(kliknout zde)

Odkazy:
Panely nástrojù

Panel 'Okno'

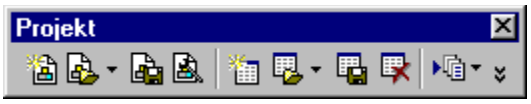
Panel nástrojù 'Okno' dává pøístup k dùležitým pøíkazùm ze stejmojmené nabídky.
Zde je obrázek panelu nástrojù. Kliknutím na jeho èásti se dozvíte více.

(kliknout zde)

Odkazy:
Panely nástrojù

Panel 'Projekt'

Panel nástrojù 'Projekt' dává pøístup k dùležitým pøíkazùm ze stejmojmené nabídky.
Zde je obrázek panelu nástrojù. Kliknutím na jeho èásti se dozvíte více.



(kliknout zde)

viz. Panely nástrojù

Projekt | Otevøít (šipka)

Zobrazí seznam nalezených skriptù. Vybraný skript otevøe v nainstalovaném nástroji pro práci se skripty.

Projekt | Spustit skript (šipka)

Zobrazí seznam nalezených skriptù. Vybraný skript spustí.

Projekt | Spustit proceduru (šipka)

Zobrazí seznam nalezených skriptù. Po vybrání jednoho otevøe dialog, do kterého lze vložit název procedury. Pak proceduru spustí.

Panel 'Dokumenty'

Panel nástrojù 'Dokumenty' dává pøístup k dùležitým pøíkazùm ze stejmojmené nabídky.
Zde je obrázek panelu nástrojù. Kliknutím na jeho èásti se dozvíte více.

(kliknout zde)

Odkazy:
Panely nástrojù

Panel 'Skript'

Panel nástrojù 'Skript' dává pøístup k dùležitým pøíkazùm ze stejmojmené nabídky.
Zde je obrázek panelu nástrojù. Kliknutím na jeho èásti se dozvíte více.

(kliknout zde)

Odkazy:
Panely nástrojù

Dokumenty

Programmer's Wizard 2 obsahuje složku *Dokumenty*, která je urèena pro pohodlný pøístup k dokumentùm, které jsou otevírány èasto a musí být vždy po ruce. Složka *Dokumenty* nemá nic spoleèného se stejnojmennou složkou ve Windows. Složka Dokumenty Windows je fyzický adresáø a obsahuje fyzické soubory zatímco složka *Dokumenty* programu Programmer's Wizard 2 je virtuální a obsahuje virtuální zástupce fyzických souborù.

Odkazy:
 Více informací o Dokumentech
 Složka Historie
 Jak zobrazit obsah složky Dokumentù
 Nabídka 'Dokumenty'
 Panel nástrojù 'Dokumenty'


Více o složce Dokumenty

Složka Dokumenty programu Programmer's Wizard 2 mùže obsahovat zástupce libovolného dokumentu. Obsah této složky mùže být zobrazen jako panel Dokumenty v hlavním oknì programu. Panel Dokumenty zobrazuje seznam všech zástupcù uvnitø složky, kliknutím na nìkterého z tìchto zástupcù se otevøe pøíslušný dokument v programu. Složka Dokumenty tak umožòuje rychlý pøístup k dokumentùm, které jsou potøeba vždy 'u ruky' kdy i proklikávání se nabídkami je zdržující.

Panel Dokumenty zobrazuje kromì dokumentù i složku Historie, která vždy obsahuje nìkolik naposledy otevøených dokumentù.

Složka Historie

Složka *Historie* vždy obsahuje nìkolik naposledy použitých dokumentù, nejvíce 16 (pro pøehlednost). Do této složky je pøidán odkaz vždy, když Programmer's Wizard 2 otevírá nìjaký dokument, lhostejno odkud. Když je použit dialog 'Otevøít', otevøen dokument z projektu nebo byl nìjaký dokument pøetažen na okno programu, vždy bude zástupce otevíraného dokumentu vložen na první místo ve složce Historie. Ke složce Historie se lze dostat dvìma zpùsoby:

Vybráním pøíkazu 'Otevøít...' z nabídky Soubor
Zobrazením složky Historie pomocí panelu Dokumenty (více..)

Odkazy:
Složka Dokumenty
Jak otevøít složku Dokumenty

Jak zobrazit složku Dokumenty

Složky Dokumenty a Historie lze zobrazit podobnì jako ostatní panely nástrojù.

1) Kliknìte pravým myšítkem na libovolný panel nástrojù
2) Z nabídky vyberte možnost *Panel 'Složka Dokumenty'*

Odkazy:
 Složka Dokumenty
 Složka Historie

Pøíruèka uživatele

Vyberte téma, o kterém se chcete dozvìdìt více.

Hledáte obecnìjší téma? Zkuste hledat v obsahu.

Automatické zálohování

Programmer's Wizard 2 obasahuje funkce pro zálohování upravovaných souborù. Program mùže být nastaven tak, aby pøed uložením upravovaného dokumentu (nezáleží na nástroji, který je použit) vytvoøil jeho záložní kopii. Mùžete si zvolit kam a jak má být záložní kopie uložena. Program mùže zmìnit jméno souboru, uložit kopii do jiné složky apod.

Nastavení této funkce je možné z dialogového okna Automatické zálohování, které se otevøe po vybrání stejnojmenného pøíkazu z nabídky Nastavení v hlavní nabídce Soubor.

Výchozí nastavení:
Po instalaci je tato funkce zcela vypnuta, po jejím spuštìní se nastaví zálohování podle klíèe *%name.~%ext*.

Odkazy:
Klíèe Automatického zálohování

Nástroje

Nástroje jsou tvoøeny moduly. Nástrojem je napøíklad SyntaxEditor nebo HexEditor.

Klíèe Automatického zálohování

Klíèe Automatického zálohování definují zpùsob, kterým probíhá zálohování. Pomocí tìchto klíèù si mùžete zvolit jméno souboru se záložní kopií.

Klíè je øetìzec znakù, který mùže obsahovat tyto makra:
%name% - název originálního dokumentu
%ext% - koncovka originálního dokumentu

Pøíklad: zálohujeme dokument s názvem "*SOUBOR.HTM*"
..pak bude makro **%name%** znamenat "*SOUBOR*"
..a makro **%ext%** bude rovno øetìzci "*HTM".*

Pomocí kombinací normálních znakù a maker lze nastavit jakékoliv jméno výsledného souboru.
Pøíklady (originálním souborem bude "*SOUBOR.HTM*"):

Klíè: "*%name%.~%ext%*"
Kopie bude mít název: "*SOUBOR.~HTM*"

Klíè: "*%name%-%ext%.tmp*"
Kopie bude mít název: "*SOUBOR-HTM.tmp*"

Specifikace cesty
Klíè mùže obsahovat i specifikaci cesty, oddìlené zpìtným lomítkem.
Pøíklad:

Klíè: "*%ext%\%name%.%ext%"*
Kopie bude vyvtoøena ve složce "*HTM*" a bude mít název "*SOUBOR.HTM*".

Odkazy:
Automatické zálohování

Klávesové zkratky

Programmer's Wizard 2 podporuje plnou nastavitelnost všech klávesových zkratek programu. Mùžete si pøizpùsobit zkratky pro jakýkoliv pøíkaz z nabídek, zmìny se projeví okamžitì po nadefinování. Tyto vlastnosti ovládání programu lze nastavit z dialogového okna *Klávesové zkratky*, které se otevøe po vybrání stejnojmenného pøíkazu z podnabídky 'Nastavení' nabídky Soubor.

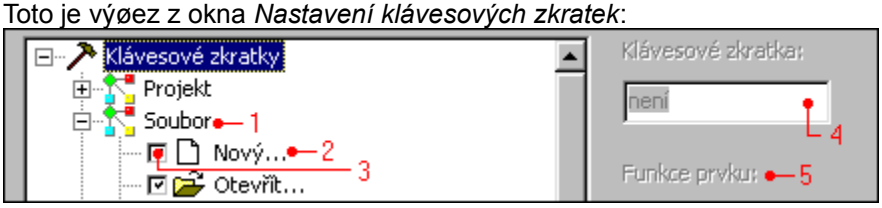
Jak na to:
1) otevøte nabídku 'Soubor' z panelu nabídek
2) vyberte nabídku 'Nastavení'
3) kliknìte na pøíkaz *Klávesové zkratky*

Zobrazí se okno *Nastavení klávesových zkratek*, hledáte-li nápovìdu k tomuto oknu, naleznete ji zde.

Odkazy:
Jak si pøizpùsobit klávesové zkratky

Jak si pøizpùsobit klávesové zkratky

Toto je výøez z okna *Nastavení klávesových zkratek*:



Legenda:
1) Skupina pøíkazù. Pøíkazy jsou dìleny do skupin podle toho, k jaké pøísluší nabídce.
2) Název a ikona pøíkazu tak, jak je vidìt v pøíslušné nabídce.
3) Zaškrtávací tlaèítko, zaškrtnutí indikuje pøítomnost klávesové zkratky.
4) Zde se zobrazuje klávesová zkratka právì vybraného pøíkazu. Pomocí tohoto pole lze také zkratku pøedefinovat. Staèí kliknout do pole a stisknout novou požadovanou klávesovou zkratku.
5) Pomocné pole, zde se zobrazí popis (krátká nápovìda) k vybranému pøíkazu.

Odkazy:
Klávesové zkratky

Externí nástroje

Programmer's Wizard 2 podporuje externí nástroje. *Externí nástroj* je libovolný program, který používáte pøi práci. Portálem k externím nástrojùm je nabídka Nástroje.

Externím nástrojem mùže být napøíklad speciální kalkulátor nebo program, který nìjak zpracuje upravovaný soubor. Mùžete si vytvoøit seznam nástrojù, které používáte a k tìm pak pøistupovat jednoduchým vybráním jehich názvu z nabídky Nástroje.

Externím nástrojùm lze pøedat pøi spuštìní nìkteré parametry, více informací naleznete pøímo v programu.

Vytvoøit seznam externích nástrojù mùžete vybráním pøíkazu 'Nastavit' z nabídky Nástroje.

Odkazy:
Nabídka 'Nástroje'

Skriptování

Tato èást nápovìdy obsahuje popis skriptovacího jazyka Pascal Script, integrovaného do prostøedí programu Programmer's Wizard 2. Je zde základní popis syntaxe jazyka, podporované datové typy a externí funkce. Je zde také nìkolik málo pøíkladù, které Vám pomohou pøed psaním vlastních skriptù.

Následující stránky **nepøedkládají** definice a vysvìtlení jazyka Pascal, to je za hranicí této nápovìdy. Naopak jsou zde popisy odlišností Pascal Scriptu od Pascalu (nebo ObjectPascalu), seznam podporovaných datových typù a seznam s vysvìtlením všech funkcí a procedur pro komunikaci s programem Programmer's Wizard 2. Znalost Pascalu pøed ètením následujících stránek je pøedpokládána.

Skriptování v programu Programmer's Wizard 2 umožòuje využívání skriptù, které za Vás budou provádìt mnoho opakovaných operací. Co je lepší než nejlepší dialogová okna pro nastavení chování programu? Skript, resp. Váš vlastní skript. Skript, který provede sérii operací a tím ušetøí Váš èas, skript, který se mùže navìsit na klíèové události programu (jako je spouštìní, naèítání modulù nebo ukonèování) a pøizpùsobit je pøesnì Vašim potøebám.

Standardní instalace programu Programmer's Wizard 2 instaluje i modul pwscript.dll, který obsahuje nástroj Editor skriptù. Tento editor s barvením syntaxe a funkcemi *Completion Proposal* a *AutoComplete* pomùže s psaním skriptù.

Odkazy:
Skript
Pascal Script

Pascal Script

Programmer's Wizard 2 má podporu skriptování v jazyce Pascal; Pascal proto, že je nejznámìjším jazykem, se kterým má zkušenosti nejširší vrstva IT veøejnosti a to hlavnì díky jeho moderní inkarnaci ObjectPascal, kterou nabízí Borland (Inprise) Delphi. Tvorba skriptu je díky Pascal Scriptu rychlá a pøehledná.
Tato verze programu nemá ještì kompletní podporu všech prvkù Pascalu, jsou zde zatím jen ty, které jsou nutné pro potøeby skriptování v programu. Další prvky budou podle potøeby pøidávány pozdìji.

Odkazy:
- Skriptování
- Datové typy
- Cykly a pøíkazy
- Matematické operace
- Knihovna funkcí a procedur
- Pøíklady

Datové typy

Pascal Script podporuje tyto datové typy:

Datový typ	Popis a rozsah
Byte	byte, neznaménkované, rozsah 0..255
ShortInt	1 byte, znaménkované, rozsah –128..127
Word	2b, neznaménkované, rozsah 0..65535
Integer	2b, znaménkované, –32768..32767
LongInt	4b, znaménkované, –2147483648..2147483647
Char	1 byte, neznaménkované, rozsah ASCII znak 0..255
String	dynamický øetìzec složený z prvkù Char
Real	8b, znaménkované, plovoucí èárka, 5.0 x 10^–324 .. 1.7 x 10^308
Single	4b, znaménkované, plovoucí èárka, 1.5 x 10^–45 .. 3.4 x 10^38
Double	8b, znaménkované, plovoucí èárka, 5.0 x 10^–324 .. 1.7 x 10^308
Extended	10b, znaménkované, plovoucí èárka, 3.6 x 10^–4951 .. 1.1 x 10^4932
Comp	8b, znaménkované, –2^63+1 .. 2^63 –1
Boolean	FALSE(0)..TRUE(1)
Array	dynamické pole z prvkù stejného typu

Nìkteré z tìchto datových typù mají svá specifika, více informací naleznete v odkazech.

Odkazy:
- Skriptování
- Pascal Script
- Matematické operace
- Cykly a pøíkazy
- Knihovna funkcí a procedur
- Pøíklady

Specifika datových typù:
- Specifika datového typu Array
- Specifika datového typu String

Skript

Zdrojový text Pascal Scriptu není pøed spuštìním pøekládán; místo toho je interpretován programem Programmer's Wizard 2 øádek po øádku pøímo za bìhu.

Specifika datového typu Array (Pascal Script)

Typ Array je polem prvkù stejného typu, pole mùže být jen jedno-rozmìrné a je dynamické, tj. mùže obsahovat libovolný poèet prvkù. Velikost pole se mìní za bìhu skriptu voláním speciální funkce. **První prvek v poli má index [0].** Deklarace dynamického pole mùže být taková:

```
var MyArray: Array of Char; //vytvoøí pole prvkù Char, poèáteèní velikost je 0
```

Dále lze ve skriptu nastavit velikost pole voláním funkce SetArrayLength:

```
//deklarace:
//procedure SetArrayLength(var Arr: Array; NewSize: LongInt);
SetArraySize(MyArray, 10); //nastavi velikost pole na 10 prvkù
```

Zjištìní velikosti pole je možné obdobnou funkcí GetArrayLength:

```
//deklarace:
//function GetArrayLength(var Arr: Array): LongInt;
ArraySize := GetArraySize(MyArray);
```

Odkazovat na jednotlivé prvky pole se lze standardnì:

```
//prohodit obsah prvního a druhého prvku pole
mTempChar := MyArray[0];
MyArray[0] := MyArray[1];
MyArray[1] := mTempChar;
```

Další odlišností pole Pascal Scriptu je jeho použití jako parametru funkcí a procedur. Pokud má být pole použito jako parametr, musí to být **vždy** parametr typu *var* a nesmí být v parametru deklarován jeho datový typ, tedy místo kompletní deklarace "*Array of String*" musí být deklarováno pole *"Array*".

```
//chybné deklarace
function SumBooleanArray(ABooleanArray: Array of Boolean): Boolean;
function SumBooleanArray(ABooleanArray: Array): Boolean;
function SumBooleanArray(var ABooleanArray: Array of Boolean): Boolean;

//správná deklarace
function SumBooleanArray(var ABooleanArray: Array): Boolean;
```

Odkazy:
- Skriptování
- Pascal Script
- Datové typy
- Cykly a pøíkazy
- Matematické operace
- Specifika datového typu String
- Knihovna funkcí a procedur
- Pøíklady

Specifika datového typu String (Pascal Script)

Datový typ String je v Pascal Scriptu dynamickým a nelze nastavit jeho délku pøi deklaraci. Nelze pøistupovat k jednotlivým prvkùm øetìzce pomocí indexování, tento pøíkaz nebude pracovat:

```
MyString[1] := 'A'; //<-- chyba zde
MyChar := MyString[1]; //<-- a zde další
```

Místo toho je nutné použít zápis využívající funkce StrGet a StrSet:

```
StrSet('A', 1, MyString);
MyChar := StrGet(MyString, 1);
```

Odkazy:
- Skriptování
- Pascal Script
- Datové typy
- Cykly a pøíkazy
- Matematické operace
- Specifika datového typu Array.
- Knihovna funkcí a procedur
- Pøíklady

Podporované pøíkazy a cykly (Pascal Script)

PascalScript podporuje v souèasné verzi tyto pøíkazy a cyklym (pøíkazy v hranatých závorkách mohou, ale nemusí být uvedeny):

```
//zakladni rozvinute vetveni
if BooleanValue then [Begin]
  Statement
[end] else [Begin]
  Statement;
[end;]

//slozeny prikaz vetveni
case OrdinalVariable of
  OrdinalValue1:
    [Begin]
      Statement;
    [end;]
  OrdinalValue2:
    [Begin]
      Statement;
    [end;]
  ...
  else
    [Begin]
      Statement;
    [end;]
end;

//cyklus FOR
for OrdinalVariable := OrdinalInitialValue {to/downto} OrdinalValue do
[Begin]
  Statement;
[end;]

//cyklus WHILE
while BooleanValue do
[Begin]
  Statement;
[end;]

//cyklus REPEAT
REPEAT
  Statement;
until BooleanValue;
```

Syntaxe tìchto pøíkazù ("if x then / else", "case x of.." "for x := y to/downto z do", "while x do" a "REPEAT until x") je zcela shodná se syntaxí jazyka Pascal, nejsou zde žádné zmìny ani omezení.

Odkazy:
- Skriptování
- Pascal Script
- Datové typy
- Matematické operace

Matematické operace (Pascal Script)

Pascal Script podporuje v souèasné verzi tyto matematicjé operátory:

+ - * /	..standardní operátory ordinálních typù (op. "+" urèen i pro øetìzce (String))
div mod	..standardní operátory dìlení
= > < <> >= <=	..porovnávací operátory (výsledkem je vždy Boolean)
and or xor not	..binární operátory

Tyto operátory se chovají stejnì jako v Pascalu.

Pozor: Pascal Script vyžaduje u nìkterých operací zavírání promìnných do závorek. Zde je seznam tìchto chybných operací a zpùsob nápravy:

Chybnì:	**Øešení:**
-OrdinalVariable	*-(OrdinalValue)*
not Variable	*not (Variable)*

Pokud bude vynechána tato závorková konvence, bude operátor "-" nebo "not" ignorován!

Toto je problém Pascal Scriptu a bude brzy odstranìn. Zde je pøíklad skriptu, který obchází tuto chybu: zde.

Odkazy:

 Pøíklad závorkové konvence "-" a "not"
 Skriptování
 Datové typy
 Cykly a pøíkazy
 Knihovna funkcí a procedur
 Pøíklady

Chyba operátorù "-" a "not" (více..) (seznam pøíkladù)
(zkopírovat do Editoru skriptù)
Tento skript demonstruje závorkovou konvenci.
{button Zkopï¿½rovat,CT()} {button Vytisknout,Print()}

```
{******************************************************}
{                                                      }
{     Programmer's Wizard 2 Pascal Script              }
{     (popis jazyka v souboru SCRIPT.DOC)              }
{                                                      }
{******************************************************}
uses ScriptUtils;

var mBooleanVariable: Boolean;
Begin
  mBooleanVariable := false;
  ShowMessage('mBooleanVariable nastaveno na FALSE.');

  mBooleanVariable := not mBooleanVariable; //poruseni zavorkove konvence
  if mBooleanVariable then Begin
    ShowMessage('Nebyl zjištìn žádný problém.');
    Halt;
  end else
    ShowMessage('Po provedení mBooleanVariable := not mBooleanVariable je ' +
      'hodnota stále FALSE.');

  mBooleanVariable := not (mBooleanVariable); //oprava zavorkove konvence
  if mBooleanVariable then
    ShowMessage('Po provedení mBooleanVariable := not (mBooleanVariable) je '
+
      'již vše v poøádku, hodnota je TRUE.')
  else //???
    ShowMessage('..ale stále pracuje funkce ShowMessage :-)');
end.
```

Minimální skript (seznam pøíkladù)
(zkopírovat do Editoru skriptù)
Toto je minimální skript, který obsahuje jen nutné èásti - fragment *Uses* a proceduru *Main*.
{button Zkopï¿½rovat,CT()} {button Vytisknout,Print()}

```
{******************************************************}
{                                                      }
{     Programmer's Wizard 2 Pascal Script              }
{     (popis jazyka v souboru SCRIPT.DOC)              }
{                                                      }
{******************************************************}
uses ScriptUtils;

Begin

end.
```

Základní skript (seznam pøíkladù)
(zkopírovat do Editoru skriptù)
Tento skript je generován Editorem skriptù pøi vytvoøení nového skriptu. Zjiš uje nainstalovanou verzi interpreta (Programmer's Wizard 2) a pokud se verze liší od požadované, zobrazí upozornìní.
{button Zkopï¿½rovat,CT()} {button Vytisknout,Print()}

```
{*******************************************************}
{                                                       }
{     Programmer's Wizard 2 Pascal Script               }
{     (popis jazyka v souboru SCRIPT.DOC)               }
{                                                       }
{*******************************************************}
uses ScriptUtils;

function GetVerStr: string;
var
  mMajor, mMinor, mRel: Integer;
Begin
  GetVersionEx(mMajor, mMinor, mRel);
  result := 'v' + IntToStr(mMajor) + '.' + IntToStr(mMinor) + '.' +
    IntToStr(mRel);
end;

function GetValidVerStr: string;
Begin
  result := 'v2.0.1'; //pozadovana verze PW
end;

Begin
  if GetValidVerStr <> GetVerStr then //nespravna verze?
    ShowMessage('Upozornìní: tento skript byl napsán pro Programmers Wizard '
+
      GetValidVerStr + '. Momentálnì spuštìná verze: ' + GetVerStr);

  //zde vlozte vas skript
end.
```

Pøíklady (Pascal Script)

Toto je seznam ukázkových skriptù, které jsou obsaženy v této nápovìdì. Kliknutím na nìkterý z nich se zobrazí jeho zdrojový text. Vkládat skripty do Editoru skriptù mùžete stiskem tlaèítka Kopírovat a následným vložením do programu pomocí funkce Vložit (nabídka nebo panel Úpravy).

Minimální skript
Základní skript
Chyba operátorù operátorù "-" a "not"

Odkazy:
Skriptování
Datové typy
Cykly a pøíkazy
Matematické operace
Knihovna funkcí a procedur

Knihovna funkcí a procedur (Pascal Script)

Zde je seznam všech funkcí a procedur, které lze používat ve skriptovacím jazyce Pascal Script programu Programmer's Wizard 2. Funkce jsou seøazeny do kategorií, uvnitø kategorií abecednì. V nìkterých kategoriích je i odkaz "Termíny použité v této kategorii"; stránka, na kterou tento link odkazuje, obsahuje dùležité informace o použitých pojmech a nemìla by být pøed použitím funkcí pøíslušné kategorie v žádném pøípadì vynechána.

Seznam kategorií:
- Obecné (vše, co není jinde, je zde)
- Pøímá komunikace s uživatelem
- Stavový øádek
- Projekt
- Plochy
- Okna
- Zásuvné moduly
- Dokumenty
- FS (soubory a složky)
- Utility a systémové operace
- Registr
- Nabídky

Odkazy:
- Skriptování
- Datové typy
- Cykly a pøíkazy
- Matematické operace
- Pøíklady

Knihovna funkcí a procedur (Pascal Script)
(zpìt do obsahu knihovny tudy)

Kategorie Obecné
EnumFiles
GetCurrentScript
GetPluginsDir
GetProjectsDir
GetScriptsDir
GetVersionEx
IsSafeScriptingEnabled
OpenFile
OpenFileEx
SendMail
SendMailEx
Sleep

Odkazy:
Skriptování
Datové typy
Cykly a pøíkazy
Matematické operace
Pøíklady

Knihovna funkcí a procedur (Pascal Script)
(zpìt do obsahu knihovny tudy)

Kategorie Pøímá komunikace s uživatelem
AskUser
InputBox
InputQuery
SelectFiles
SelectDirectory
ShowCheckListDialog
ShowListDialog
ShowMessage

Odkazy:
Skriptování
Datové typy
Cykly a pøíkazy
Matematické operace
Pøíklady

Knihovna funkcí a procedur (Pascal Script)
(zpìt do obsahu knihovny tudy)

Kategorie Stavový øádek
CancelStatusBarMessage
GetProgressBarStatus
ResetProgressBar
SetProgressBarStatus
SetStatusBarMessage

Odkazy:
Skriptování
Datové typy
Cykly a pøíkazy
Matematické operace
Pøíklady

Knihovna funkcí a procedur (Pascal Script)
(zpìt do obsahu knihovny tudy)

Termíny použíté v této kategorii
Kategorie Projekt
CloseProject
CreateProject
EnumProjectFiles
GetCurrentProject
GetProjectCount
GetProjectInfo
IsProjectOpened
OpenProject
SetProjectFiles

Odkazy:
Skriptování
Datové typy
Cykly a pøíkazy
Matematické operace
Pøíklady

Knihovna funkcí a procedur (Pascal Script)
(zpìt do obsahu knihovny tudy)

Termíny použité v této kategorii
Kategorie Plochy
CloseWorkspace
CreateWorkspace
CreateWorkspaceEx
GetCurrentWorkspace
GetWorkspaceCount
GetWorkspaceCountEx
GetWorkspaceFile
GetWorkspaceFileEx
GetWorkspaceInfo
GetWorkspaceInfoEx
OpenWorkspace

Odkazy:
Skriptování
Datové typy
Cykly a pøíkazy
Matematické operace
Pøíklady

Knihovna funkcí a procedur (Pascal Script)
(zpìt do obsahu knihovny tudy)

Kategorie FS (soubory a složky)
CopyFile
DeleteFile
ExecuteFile
ExtractFileExt
ExtractFileName
ExtractFilePath
FileExists
RenameFile

Odkazy:
Skriptování
Datové typy
Cykly a pøíkazy
Matematické operace
Pøíklady

Knihovna funkcí a procedur (Pascal Script)
(zpìt do obsahu knihovny tudy)

Kategorie Utility
Copy
Cos
Dec
Delete
GetArrayLength
Halt
Chr
Inc
Insert
IntToStr
Length
LowerCase
Ord
Pi
Pos
Random
Randomize
Round
SetArrayLenght
Sin
StrGet
StrSet
StrToInt
Trunc
UpperCase

Odkazy:
Skriptování
Datové typy
Cykly a pøíkazy
Matematické operace
Pøíklady

Knihovna funkcí a procedur (Pascal Script)
(zpìt do obsahu knihovny tudy)

Kategorie Registr
RegReadInt
RegReadString
RegWriteInt
RegWriteString

Odkazy:
Skriptování
Datové typy
Cykly a pøíkazy
Matematické operace
Pøíklady

Knihovna funkcí a procedur (Pascal Script)
(zpìt do obsahu knihovny tudy)

Kategorie **Nabídky** má odlišnou strukturu od ostatních, nejsou zde detailní popisy funkcí. Všechny funkce v této kategorii jsou urèeny pro simulaci kliknutí myšítkem do nabídek programu. Každému jednotlivému pøíkazu z nabídek odpovídá jedna konkrétní funkce.

Tyto funkce mají shodnou deklaraci:

function Menu_x: Boolean;

Návratová hodnota:
Funkce vrací *TRUE* pokud byl pøíkaz úspìšnì simulován, jinak vrací *FALSE*.

- Menu_FormatDecIndent
- Menu_FormatIncIndent
- Menu_FormatItalic
- Menu_FormatLeftAlign
- Menu_FormatNoAlign
- Menu_FormatNumBullets
- Menu_FormatRightAlign
- Menu_FormatTextColor
- Menu_FormatUnderline
- Menu_HelpAbout
- Menu_HelpRegister
- Menu_HelpTip
- Menu_HelpWWW
- Menu_ProjectClose
- Menu_ProjectDeleteWorkspace
- Menu_ProjectNew
- Menu_ProjectNewWorkspace
- Menu_ProjectOpen
- Menu_ProjectOpenWorkspace
- Menu_ProjectSave
- Menu_ProjectSaveWorkspace
- Menu_ProjectSetup
- Menu_ScriptExecute
- Menu_ScriptExecuteProc
- Menu_ScriptOpenExecute
- Menu_ScriptOpenExecuteProc
- Menu_ScriptNew
- Menu_ScriptOpen
- Menu_ToolsSetup
- Menu_ViewDesktopOptions
- Menu_ViewFullscreen
- Menu_ViewStatusBar
- Menu_WindowArrange
- Menu_WindowArrangeIcons
- Menu_WindowArrangeV
- Menu_WindowCascade
- Menu_WindowClose
- Menu_WindowCloseAll
- Menu_WindowMinimize
- Menu_WindowNext
- Menu_WindowPrevious

Odkazy:

- Skriptování
- Datové typy
- Cykly a pøíkazy
- Matematické operace
- Pøíklady

Knihovna funkcí - prohlížení

Kategorie: *Obecné*
Téma: OpenFile

Deklarace:
function OpenFile(FileName: String): Boolean

Parametry:
FileName: String
Název dokumentu k otevøení

Popis:
Funkce *OpenFile* otevírá dokument udaný parametrem *FileName* v programu Programmer's Wizard 2. Dokument se otevøe ve výchozím nástroji tak, jako by byl otevøen dialogem Otevøít a bude pøidán do složky Historie. Pokud je dokument již otevøený, pøepne se jeho okno do popøedí. Tato funkce vždy otevøe soubor v programu Programmer's Wizard 2, pro spuštìní programu nebo otevøení dokumentu jeho asociovaným program je vhodnìjší funkce *ExecuteFile*.

Návratový výraz:
Funkce vrací *TRUE* pokud byl soubor úspìšnì otevøen, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: Pøímá komunikace s uživatelem
Téma: AskUser

Deklarace:
function AskUser(Question: String): Boolean

Parametry:
Question: String
Dotaz na uživatele

Popis:
Funkce *AskUser* zobrazuje dotazovací dialog. Parametr *Question* musí obsahovat otázku, které má být zobrazena uživateli, uživatel pak mùže vybrat odpovìï *Ano* nebo *Ne*. Odpovìï uživatele vrací funkce jako typ Boolean.

Návratový výraz:
Funkce vrací *TRUE* pokud bylo stisknuto tlaèítko Ano, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: *Pøímá komunikace s uživatelem*
Téma: InputBox

Deklarace:
function InputBox(Caption: String; Prompt: String; Default: String): String

Parametry:
Caption: String
Titulek okna
Prompt: String
Text uvnitø okna
Default: String
Text v editaèním poli po zobrazení dialogu

Popis:
Funkce *InputBox* zobrazuje dialog s dvìma tlaèítky (*OK* a *Storno*) a jedním editaèním polem, do kterého lze zadávat text. Okno bude mít nadpis definovaný parametrem *Caption*, popisek editaèního pole bude nastaven na parametr *Prompt*. Editaèní pole bude obsahovat text udaný parametrem *Default*. Funkce vrací obsah editaèního pole po uzavøení okna.

Návratový výraz:
Funkce vrací vždy obsah editaèního pole, nezávisle na tom, zda uživatel stiskl OK nebo Storno. Pro pøípady kde je nutné znát i zpùsob, jakým bylo okno uzavøeno (jestli tlaèítkem *OK* nebo *Storno*), je vhodnìjší funkce InputQuery.

Knihovna funkcí - prohlížení

Kategorie: Pøímá komunikace s uživatelem
Téma: InputQuery

Deklarace:
function InputQuery(Caption: String; Prompt: String; var Default: String): Boolean

Parametry:
Caption: String
Titulek okna
Prompt: String
Text uvnitø okna
var Default: String
Text v editaèním poli po zobrazení dialogu

Popis:
Funkce *InputQuery* zobrazuje dialog s dvìma tlaèítky (*OK* a *Storno*) a jedním editaèním polem, do kterého lze zadávat text. Okno bude mít nadpis definovaný parametrem *Caption*, popisek editaèního pole bude nastaven na parametr *Prompt*. Editaèní pole bude obsahovat text udaný parametrem *Default*. Funkce vrací obsah editaèního pole po uzavøení okna v parametru *Default*.

Návratový výraz:
Funkce vrací *TRUE* pokud bylo okno zavøeno stiskem tlaèítka *OK*, pøi stisku *Storno* vrací *FALSE*. Obsah editaèního pole po zavøení okna je vrácen do parametru *Default*. Pro pøípady kde není nutné znát zpùsob, jakým bylo okno uzavøeno (jestli tlaèítkem *OK* nebo *Storno*), je vhodnìjší funkce InputBox.

Knihovna funkcí - prohlížení

Kategorie: Pøímá komunikace s uživatelem
Téma: ShowListDialog

Deklarace:
function ShowListDialog(Caption, Prompt: String; var Lines: Array of string): LongInt

Parametry:
Caption: String
Titulek okna
Prompt: String
Text uvnitø okna
var Lines: Array of string
Seznam øetìzcù, které se mají zobrazit jako jednotlivé položky

Popis:
Funkce *ShowListDialog* umožòuje uživateli vybrat si ze seznamu øetìzcových výrazù. Funkce zobrazuje okno s nadpisem udaným parametrem *Caption*. Okno obsahuje tlaèítka *OK* a *Storno*, také obsahuje pole se seznamem øetìzcù, ze kterých mùže uživatel jeden vybrat a stisknout tlaèítko *OK*. Obsah seznamu je definován parametrem *Lines*. Funkce vrací index vybraného øetìzce.

Návratový výraz:
Funkce vrací index vybraného øetìzce. Tento index lze pøímo použít do pole *Lines* takto: "Lines[result]" kde *result* je hodnota vrácená funkcí ShowListDialog. Když uživatel zruší výbìr prvku tlaèítkem *Storno*, funkce vrací hodnotu -1. Skript musí vždy ovìøit, zda není výsledná hodnota -1 protože bez tohoto testu by mohla být provedena tato instrukce: "Lines[-1]". Instrukce je neplatná protože první prvek z pole má index 0 - skript by byl okamžitì ukonèen s chybovým hlášením.

Knihovna funkcí - prohlížení

Kategorie: Pøímá komunikace s uživatelem
Téma: ShowMessage

Deklarace:
procedure ShowMessage(Message: String)

Parametry:
Message: String
Informace pro uživatele

Popis:
Funkce *ShowMessage* zobrazuje informaèní okno s titulkem *Programmer's Wizard 2* a textem uvnitø okna definovaným parametrem *Message*. Okno má jediné tlaèítko; tlaèítko *OK*. Po jeho stisku se okno uzavøe a skript mùže pokraèovat.

Návratový výraz:
Nevrací žádný výraz.

Knihovna funkcí - prohlížení

Kategorie: *Stavový øádek*
Téma: CancelStatusBarMessage

Deklarace:
procedure CancelStatusBarMessage

Popis:
Funkce *CancelStatusBarMessage* ruší zpávu na stavovém øádku vytvoøenou funkcí SetStatusBarMessage. Skript by mìl zavolat tuto funkci vždy pøed ukonèením, pokud tyto funkce používal a nemá jiný dùvod pro zanechání své zprávy na stavovém øádku.

Poznámka:
Jako bezpeèností opatøení proti skriptùm které po sobì neodstraní zprávu na stavovém øádku je zpráva vytvoøená pomocí funkce SetStatusBarMessage omezená èasem 10 vteøin. Po uplynutí této doby bude zpráva odstranìna automaticky.

Návratový výraz:
Nevrací žádný výraz.

Knihovna funkcí - prohlížení

Kategorie: *Stavový øádek*
Téma: GetProgressBarStatus

Deklarace:
procedure GetProgressBarStatus(var Position: LongInt; var Maximum: LongInt)

Parametry:
var Position: LongInt
Aktuální hodnota ukazatele prùbìhu
var Maximum: LongInt
Maximální hodnota ukazatele prùbìhu

Popis:
Funkce *GetProgressBarStatus* vrací aktuální stav ukazatele prùbìhu. Tento ukazatel se nachází na stavovém øádku a je sdílen jak skripty, tak nástroji. Ukazatel lze nastavit obdobnou funkcí SetProgressBarStatus.

Návratový výraz:
Funkce vrací aktuální a maximální hodnotu ukazatele prùbìhu pomocí parametrù *Position* a *Maximum*. Pokud se obì hodnoty rovnají nule (0), není ukazatel zobrazen.

Návratový výraz:
Nevrací žádný výraz.

Knihovna funkcí - prohlížení

Kategorie: *Stavový øádek*
Téma: ResetProgressBar

Deklarace:
procedure ResetProgressBar

Popis:
Funkce *ResetProgressBar* nastavuje stavový øádek zpìt do klidového stavu. Ukazatel prùbìhu bude skryt a jeho hodnoty *Position* a *Maximum* budou nastaveny na nulu (0). Skript by mìl zavolat tuto funkci pøed svým ukonèením vždy pøed svým ukonèením, pokud ukazatel používal, jinak zùstane ukazatel zobrazen i po ukonèení skriptu.

Návratový výraz:
Nevrací žádný výraz.

Návratový výraz:
Nevrací žádný výraz.

Poznámka:
Jako bezpeèností opatøení proti skriptùm, které po sobì neodstraní ukazatel prùbìhu, je ukazatel vytvoøený pomocí funkce SetProgressBarStatus omezen èasem 8 vteøin. Pokud se stav ukazatele prùbìhu za tuto dobu nezmìní (nezáleží jak; technicky by však mìlo jít o pøiblížení hodnoty *Position* parametru *Maximum*), bude ukazatel odstranìn automaticky.

Knihovna funkcí - prohlížení

Kategorie: *Stavový øádek*
Téma: SetStatusBarMessage

Deklarace:
procedure SetStatusBarMessage(Message: String)

Parametry:
Message: String
Nová zpráva pro zobrazení na stavovém øádku

Popis:
Funkce *SetStatusBarMessage* nastavuje zprávu na stavovém øádku. Všechny druhotné panely na stavovém øádku budou ukryty a celý jeho prostor bude vyhrazen zprávì skriptu. Ta bude vykreslena tuèným písmem tak, aby byla kontrastní k ostatním èástem okna. Zpráva bude zobrazena až do zavolání funkce CancelStatusBarMessage.

Návratový výraz:
Nevrací žádný váraz.

Poznámka:
Jako bezpeèností opatøení proti skriptùm které po sobì neodstraní zprávu na stavovém øádku je zpráva vytvoøená pomocí funkce SetStatusBarMessage omezená èasem 10 vteøin. Po uplynutí této doby bude zpráva odstranìna automaticky.

Knihovna funkcí - prohlížení

Kategorie: Stavový øádek
Téma: SetProgressBarStatus

Deklarace:
procedure SetProgressBarStatus(Position, Maximum: LongInt)

Parametry:
Position: LongInt
Aktuální hodnota ukazatele prùbìhu
Maximum: LongInt
Maximální hodnota ukazatele prùbìhu

Popis:
Funkce *SetProgressBarStatus* nastavuje sdílený ukazatel prùbìhu. Tento ukazatel se zobrazuje na stavovém øádku. Vykreslován je jako standardní ukazatel Windows. Parametry *Position* a *Maximum* jsou interpretovány jako "*Position* hotovo
z *Maximum* celkem", tedy skript by mìl nastavit hodnotu *Position* na nulu (0) pøi zaèátku dlouhé operace a postupnì ji zvyšovat tak, aby pøi dokonèení byla rovna hodnotì *Maximum*. Tento ukazatel je nutné po dokonèení operaci odstranit voláním funkce ResetProgressBar. Skript by mìl tento ukazatel používat pøi provádìní èasovì nároèných operací.

Návratový výraz:
Nevrací žádný výraz.

Poznámka:
Jako bezpeèností opatøení proti skriptùm, které po sobì neodstraní ukazatel prùbìhu, je ukazatel vytvoøený pomocí funkce SetProgressBarStatus omezen èasem 8 vteøin. Pokud se stav ukazatele prùbìhu za tuto dobu nezmìní (nezáleží jak; technicky by však mìlo jít o pøiblížení hodnoty *Position* parametru *Maximum*), bude ukazatel odstranìn automaticky.

Knihovna funkcí - prohlížení

Kategorie: *Projekt*
Téma: CloseProject

Deklarace:
function CloseProject: Boolean

Popis:
Funkce *CloseProject* uzavírá projekt, pokud je nìjaký otevøen. Zavøení projektu zpùsobí zavøení plochy a tím i všech otevøených oken. Uživateli bude dána možnost uložit zmìny, pokud nìjaké existují. Když uživatel neuloží svou práci a místo toho stiskne tlaèítko *Storno*, nebude nìkteré z oken uzavøeno, neuzavøe se plocha a ani projekt; v takovém pøípadì funkce vrací *FALSE*. Po provedení této funkce s výsledkem *TRUE* lze pøedpokládat, že je otevøeny základní plocha programu a není otevøen jakýkoliv dokument.

Návratový výraz:
Funkce vrací *TRUE* pokud byl úspìšnì zavøen projekt, jeho plocha s ní i všechna otevøená okna. Funkce vrací *TRUE* i tehdy, pokud nebyl žádný projekt otevøen; v takovém pøípadì jednoduše uzavøe všechna okna a plochu.

Knihovna funkcí - prohlížení

Kategorie: *Projekt*
Téma: GetCurrentProject

Deklarace:
function GetCurrentProject: LongInt

Popis:
Funkce *GetCurrentProject* vrací ID právì otevøeného projektu. Toto ID mùže být použito v dalších funkcích této kategorie. ID je garantováno jako hodnota rozsahu *0* <= *ID* < *GetProjectCount* a je indexem do vnitøní tabulky projektù programu. Skript by mìl pøed voláním této funkce vždy napøed zjistit, zda je nìjaký projekt otevøen a to funkcí *IsProjectOpened*.

Návratový výraz:
Funkce vrací vždy ID právì otevøeného projektu. Pokud není žádný projekt otevøen, vrací funkce hodnotu *-1*, což je pro všechny další funkce neplatné ID.

Knihovna funkcí - prohlížení

Kategorie: *Projekt*
Téma: GetProjectCount

Deklarace:
function GetProjectCount: LongInt

Popis:
Funkce *GetProjectCount* vrací poèet existujících projektù. Tuto informaci lze použít pro procházení projektù, protože ID projektu je garantováno jako hodnota rozsahu *0* <= *ID* < *GetProjectCount* a je indexem do vnitøní tabulky projektù programu.

Návratový výraz:
Funkce vrací vždy poèet existujících projektù.

Knihovna funkcí - prohlížení

Kategorie: *Projekt*
Téma: GetProjectCount

Deklarace:
function GetProjectInfo(ProjectID: LongInt; var FileName, Name, Description: string): Boolean;

Parametry:
ProjectID: LongInt
ID projektu
var FileName: string
Do tohoto parametru je zkopírován název souboru s projektem
var Name: string
Do tohoto parametru je zkopírován název projektu
var Description: string
Do tohoto parametru je zkopírován popis projektu, pokud byl nìjaký zadán

Popis:
Funkce *GetProjectInfo* se používá ke zjištìní informací o projektu. Projekt je identifikován parametrem *ProjectID*, informace o projektu jsou zkopírovány do parametrù *FileName*, *Name* a *Description*. *ProjectID* musí být v rozsahu *0* <= *ProjectID* < *GetProjectCount*.

Návratový výraz:
Funkce vrací *TRUE* pokud byly informace zkopírovány do parametrù, jinak vrací *FALSE*. Funkce vìtšinou vrátí *FALSE* jako následek nesprávného parametru *ProjectID*.

Knihovna funkcí - prohlížení

Kategorie: *Projekt*
Téma: IsProjectOpened

Deklarace:
function IsProjectOpened: Boolean

Popis:
Funkci *IsProjectOpened* lze použít pro získání informace, zda je právì otevøen nìjaký projekt. ID otevøeného projektu lze zjistit voláním funkce *GetCurrentProject* .Skript by pøed voláním *GetCurrentProject* vždy mìl zavolat napøed tuto funkci a ujistit se, že je nìjaký projekt otevøen.

Návratový výraz:
Funkce vrací *TRUE* pokud je v dobì jejího volání otevøen nìjaký projekt, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: Projekt
Téma: OpenProject

Deklarace:
function OpenProject(ProjectID: LongInt): Boolean

Parametry:
ProjectID: LongInt
ID projektu, který má být otevøen

Popis:
Funkce *OpenProject* otevírá projekt s ID udaným parametrem *ProjectID*. Pokud je již nìjaký projekt otevøen, funkce se ho napøed pokusí zavøít, mùže také zobrazit dialog s upozornìním pro uživatele na zmìny v dokumentech, které se zavírají a které je nutné uložit, funkce prostì udìlá vše co funkce *CloseProject a* teprve pak projekt otevøe.

Návratový výraz:
Funkce vrací *TRUE* pokud byl nový projekt úspìšnì otevøen, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: *Plochy*
Téma: CloseWorkspace

Deklarace:
function CloseWorkspace: Boolean

Popis:
Funkce *CloseWorkspace* uzavírá plochu právì otevøeného projektu, pokud je nìjaký otevøen. Zavøení plochy zpùsobí i zavøení všech otevøených oken. Uživateli bude dána možnost uložit zmìny, pokud nìjaké existují. Když uživatel neuloží svou práci a místo toho stiskne tlaèítko *Storno*, nebude nìkteré z oken uzavøeno, neuzavøe se plocha; v takovém pøípadì funkce vrací *FALSE*. Po provedení této funkce s výsledkem *TRUE* lze pøedpokládat, že není otevøen jakýkoliv dokument.

Návratový výraz:
Funkce vrací *TRUE* pokud byla úspìšnì zavøena plocha a s ní i všechna otevøená okna. Funkce vrací *TRUE* i tehdy, pokud nebyla žádná plocha otevøena; v takovém pøípadì jednoduše uzavøe všechna okna.

Knihovna funkcí - prohlížení

Kategorie: *Plochy*
Téma: GetCurrentWorkspace

Deklarace:
function GetCurrentWorkspace: LongInt

Popis:
Funkce *GetCurrentWorkspace* vrací ID právì otevøené plochy otevøeného projektu, pokud je nìjaký otevøen. Toto ID mùže být použito v dalších funkcích této kategorie. ID je garantováno jako hodnota rozsahu *0* <= *ID* < *GetWorkspaceCount* a je indexem do vnitøní tabulky ploch programu. Skript by mìl pøed voláním této funkce vždy napøed zjistit, zda je nìjaký projekt otevøen a to funkcí *IsProjectOpened*.

Návratový výraz:
Funkce vrací vždy ID právì otevøené plochy. Pokud není žádný projekt otevøen (resp. není otevøena žádní plocha), vrací funkce hodnotu *-1*, což je pro všechny další funkce neplatné ID.

Knihovna funkcí - prohlížení

Kategorie: *Plochy*
Téma: GetWorkspaceCount

Deklarace:
function GetWorkspaceCount: LongInt

Popis:
Funkce *GetWorkspaceCount* zjiš□uje poèet ploch v právì otevøeném projektu. Vrácenou hodnotu lze použít pro procházení plochami, protože identifikátor plochy WorkspaceID je garantován jako hodnota *0* <= *WorkspaceID* < *GetWorkspaceCount*. Skript by mìl pøed voláním této funkce vždy napøed zjistit, zda je nìjaký projekt otevøen a to funkcí *IsProjectOpened*.

Návratový výraz:
Funkce vrací vždy poèet ploch v otevøebém projektu, pokud není otevøen žádný projekt, vrací hodnotu 0.

Knihovna funkcí - prohlížení

Kategorie: *Plochy*
Téma: GetWorkspaceCountEx

Deklarace:
function GetWorkspaceCountEx(ProjectID: LongInt): LongInt

Parametry:
ProjectID: LongInt
ID projektu, na který je požadavek smìrován

Popis:
Funkce *GetWorkspaceCountEx* pracuje stejnì jako funkce *GetWorkspaceCount* s tím rozdílem, že mùže zjistit poèet ploch jakéhokoliv, tedy ne jen právì otevøeného projektu. Více informací lze nalézt v popisu funkce GetWorkspaceCount.

Návratový výraz:
Funkce vrací vždy poèet ploch v projektu definovaném parametrem *ProjectID*, pøi neúspìchu vrací hodnotu 0.

Knihovna funkcí - prohlížení

Kategorie: *Plochy*
Téma: GetWorkspaceFile

Deklarace:
function GetWorkspaceFile(WorkspaceID, FileID: LongInt): String

Parametry:
WorkspaceID: LongInt
ID plochy, na kterou je požadavek smìrován
FileID: LongInt
ID dokumentu, jehož plnou cestu má funkce vrátit

Popis:
Funkce *GetWorkspaceFile* zjiš uje plnou cestu *FileID*-tého dokumentu na ploše s ID definovaným parametrem *WorkspaceID* právì otevøeného projektu. Parametr *FileID* musí být hodnota v rozsahu *0* <= *FileID* < *poèet dokumentù na ploše*. Poèet dokumentù na ploše lze zjistit voláním funkce *GetWorkspaceInfo*.

Návratový výraz:
Funkce vrací plnou cestu k vyžádanému dokumentu, pokud je parametr *WorkspaceID* nebo *FileID* neplatný, vrací prázdný øetìzec.

Knihovna funkcí - prohlížení

Kategorie: Plochy
Téma: GetWorkspaceFileEx

Deklarace:
function GetWorkspaceFileEx(ProjectID, WorkspaceID, FileID: LongInt): String

Parametry:
ProjectID: LongInt
ID projektu, jehož plocha má být definovaná parametrem *WorkspaceID*
WorkspaceID: LongInt
ID plochy, na kterou je požadavek smìrován
FileID: LongInt
ID dokumentu, jehož plnou cestu má funkce vrátit

Popis:
Funkce *GetWorkspaceFileEx* zjiš uje plnou cestu *FileID*-tého dokumentu na ploše s ID definovaným parametrem *WorkspaceID* v projektu *ProjectID*. Parametr *FileID* musí být hodnota v rozsahu *0* <= *FileID* < *poèet dokumentù na ploše*. Poèet dokumentù na ploše lze zjistit voláním funkce *GetWorkspaceInfoEx*.

Návratový výraz:
Funkce vrací plnou cestu k vyžádanému dokumentu, pokud je nìkterý z parametrù neplatný, vrací prázdný øetìzec.

Knihovna funkcí - prohlížení

Kategorie: Plochy
Téma: GetWorkspaceInfo

Deklarace:
function GetWorkspaceInfo(WorkspaceID: LongInt; var Name, Description: String; var FileCount: LongInt): Boolean

Parametry:
WorkspaceID: LongInt
ID plochy, na kterou je požadavek smìrován
var Name: String
Do tohoto parametru je vrácen název plochy
var Description: String
Do tohoto parametru je vrácen název plochy, pokud nìjaký existuje
var FileCount: LongInt
Do tohoto parametru je vrácen poèet otevøených dokumentù na ploše

Popis:
Funkce *GetWorkspaceInfo* se používá ke zjištìní informací o nìkteré z ploch právì otevøeného projektu. Právì otevøenou plochu lze zjistit voláním funkce *GetCurrentWorkspace*. Procházet plochy lze voláním *GetWorkspaceCount* a následným procházením ID ploch podle klíèe "*for WorkspaceID := 0 to GetWorkspaceCount - 1 do...".* Funkce vrací název a popis plochy, vrací také poèet otevøených dokumentù. Cesty k jednotlivým dokumentùm lze zjistit funkcí GetWorkspaceFile.

Návratový výraz:
Funkce vrací *TRUE* pokud vše probìhlo v poøádku a údaje byly zkopírovány do parametrù, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: Plochy
Téma: GetWorkspaceInfoEx

Deklarace:
function GetWorkspaceInfoEx(ProjectID, WorkspaceID: LongInt; var Name, Description: String; var FileCount: LongInt): Boolean

Parametry:
ProjectID: LongInt
ID projektu, jehož plocha je definována parametrem *WorkspaceID*
WorkspaceID: LongInt
ID plochy, na kterou je požadavek smìrován
var Name: String
Do tohoto parametru je vrácen název plochy
var Description: String
Do tohoto parametru je vrácen název plochy, pokud nìjaký existuje
var FileCount: LongInt
Do tohoto parametru je vrácen poèet otevøených dokumentù na ploše

Popis:
Funkce *GetWorkspaceInfo* se používá ke zjištìní informací o nìkteré z ploch projektu definovaného parametrem *ProjectID*. Procházet plochy lze voláním *GetWorkspaceCountEx* a následným procházením ID ploch podle klíèe "*for WorkspaceID := 0 to GetWorkspaceCountEx(ProjectID) - 1 do...".* Funkce vrací název a popis plochy, vrací také poèet otevøených dokumentù. Cesty k jednotlivým dokumentùm lze zjistit funkcí *GetWorkspaceFileEx*.

Návratový výraz:
Funkce vrací *TRUE* pokud vše probìhlo v poøádku a údaje byly zkopírovány do parametrù, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: Plochy
Téma: OpenWorkspace

Deklarace:
function OpenWorkspace(WorkspaceID: LongInt): Boolean

Parametry:
WorkspaceID: LongInt
ID plochy, která se má otevøít

Popis:
Funkce *OpenWorkspace* otevírá plochu právì otevøeného projektu. Plocha je urèena parametrem *WorkspaceID*. Pøi provádìní této funkce se napøed uzavøe existující plocha a s ní i všechny otevøené dokumenty, uživateli bude dána možnost uložit zmìny v dokumentech, pokud nìjaké existují. Po uzavøení všech oken a plochy se otevøe plocha *WorkspaceID*.

Návratový výraz:
Funkce vrací *TRUE* pokud byla plocha úspìšnì otevøena, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: FS (soubory a složky)
Téma: DeleteFile

Deklarace:
function DeleteFile(FileName: String): Boolean

Parametry:
FileName: String
Plná cesta k souboru, který se má smazat

Popis:
Funkce *DeleteFile* maže fyzický soubor udaný parametrem *FileName*.

Návratový výraz:
Funkce vrací *TRUE* pokud byl soubor smazán, jinak vrací *FALSE*.

Poznámky:
Tato funkce je jednou z chránìných funkcí a mùže být v nastavení skriptování zakázána. Skript by si mìl vždy ovìøit zda jsou tyto funkce povoleny voláním *IsSafeScriptingEnabled*.

Knihovna funkcí - prohlížení

Kategorie: FS (soubory a složky)
Téma: ExecuteFile

Deklarace:
function ExecuteFile(FileName: String; Silent: Boolean): Boolean

Parametry:
FileName: String
Plná cesta k souboru, který se má spustit
Silent: Boolean
Udává, zda má funkce pøi neúspìchu jen vrátit *FALSE*, nebo zobrazit upozornìní.

Popis:
Funkce Execute*File* spustí program nebo otevøe dokument udaný parametrem *FileName*. Pokud se jedná o doukment, bude tento otevøen ve svém asociovaném programu, ne v programu Programmer's Wizard 2 jak by se stalo pøi použití funkce *OpenFile*. Pokud je parametr *Silent* nastaven na *FALSE* a program nebo dokument se nepodaøí spustit, zobrazí funkce i upozornìní viditelné uživateli.

Návratový výraz:
Funkce vrací *TRUE* pokud byl program nebo dokument spuštìn, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: *FS (soubory a složky)*
Téma: ExtractFileExt

Deklarace:
function ExtractFileExt(FileName: String): String

Parametry:
FileName: String
Plná cesta k souboru, jehož koncovka se má zjistit

Popis:
Funkce *ExtractFileExt* získává z názvu souboru udaného parametrem *FileName* jeho koncovku a tu vrací jako øetìzec. ("C:\SOUBOR.HTM" bude pøevedeno na ".HTM")

Návratový výraz:
Funkce vždy vrací koncovku souboru udaného parametrem *FileName*.

Knihovna funkcí - prohlížení

Kategorie: FS (soubory a složky)
Téma: ExtractFileName

Deklarace:
function ExtractFileName(FileName: String): String

Parametry:
FileName: String
Plná cesta k souboru, jehož jméno se má zjistit

Popis:
Funkce *ExtractFileName* získává z plné cesty souboru udaného parametrem *FileName* jeho název s koncovkou a výsledek vrací jako øetìzec. ("C:\SOUBOR.HTM" bude pøevedeno na "SOUBOR.HTM")

Návratový výraz:
Funkce vždy vrací název souboru udaného parametrem *FileName*.

Knihovna funkcí - prohlížení

Kategorie: FS (soubory a složky)
Téma: ExtractFilePath

Deklarace:
function ExtractFilePath(FileName: String): String

Parametry:
FileName: String
Plná cesta k souboru, jehož cesta se má zjistit

Popis:
Funkce *ExtractFileName* získává z plné cesty a názvu souboru udaného parametrem *FileName* plnou cestu a výsledek vrací jako øetìzec. ("C:\SOUBOR.HTM" bude pøevedeno na "C:\")

Návratový výraz:
Funkce vždy vrací název souboru udaného parametrem *FileName*.

Knihovna funkcí - prohlížení

Kategorie: *FS (soubory a složky)*
Téma: FileExists

Deklarace:
function FileExists(FileName: String): Boolean

Parametry:
FileName: String
Plná cesta k souboru, který se má ovìøit

Popis:
Funkce *FileExists* zjiš uje, jestli soubor udaný parametrem *FileName* existuje a vrací *TRUE* pro existující soubor, *FALSE* pro neexistující.

Návratový výraz:
Funkce vrací *TRUE* pokud byl soubor udaný parametrem *FileName* existuje, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: *FS (soubory a složky)*
Téma: RenameFile

Deklarace:
function RenameFile(OldName, NewName: String): Boolean

Parametry:
OldName: String
Plná cesta k souboru, který má být pøejmenován
NewName: String
Nový název souboru

Popis:
Funkce Rename*File* pøejmenuje fyzický soubor udaný parametrem *OldName* na název *NewName*.

Návratový výraz:
Funkce vrací *TRUE* pokud byl soubor smazán, jinak vrací *FALSE*.

Poznámky:
Tato funkce je jednou z chránìných funkcí a mùže být v nastavení skriptování zakázána. Skript by si mìl vždy ovìøit zda jsou tyto funkce povoleny voláním *IsSafeScriptingEnabled*.

Knihovna funkcí - prohlížení

Kategorie: Obecné
Téma: IsSafeScriptingEnabled

Deklarace:
function IsSafeScriptingEnabled: Boolean

Popis:
Funkce *IsSafeScriptingEnabled* zjiš□uje stav skriptování a vrací *TRUE*, pokud skript bìží pøi zapnutém bezpeèném skriptování. Bezpeèné skriptování lze zapnout a vypnout z dialogu *Nastavení skriptování*, které se zobrazí po vybrání stejnojmenného pøíkazu z podnabídky *Nastavení* nabídky *Soubor*. Pøi zapnutém bezpeèném skriptování nelze provádìt funkce jako jsou *DeleteFile* a *RenameFile*.

Návratový výraz:
Funkce vrací *TRUE* pokud je bezpeèné skriptování povoleno, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Copy

Deklarace:
function Copy(S: String; Index, Legth: LongInt): String

Parametry:
S: String
Zdrojový øetìzec
Index: LongInt
Index prvního znaku (první v øetìzci je index 1)
Length: LongInt
Délka výsledného øetìzce

Popis:
Funkce *Copy* kopíruje *Length* znakù z øetìzce *S*, prvním znakem je znak s indexem udaným parametrem *Index*. Výsledek vrací jako øetìzec.

Návratový výraz:
Funkce vrací *Length* znakù z øetìzce *S*, prvním znakem je znak na pozici *Index*.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Cos

Deklarace:
function Cos(X: Extended): Extended

Parametry:
X: Extended
Parametr funkce Cos

Popis:
Funkce provádí standardní operaci *Cos*, parametrem je *X*. Parametr i výsledek jsou typu *Extended* a jsou chápány v radiánech.

Návratový výraz:
Funkce vždy vrací výsledek operace *Cos X*.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Dec

Deklarace:
procedure Dec(var X: LongInt);

Parametry:
var X: LongInt
Promìnná, jejíž obsah má být dekrementován.

Popis:
Funkce *Dec* snižuje hodnotu promìnné *X* o jednu a je ekvivalentní zápisu "*X := X - 1*".

Návratový výraz:
Nevrací žádný výraz pøímo, mìní hodnotu parametru *X*.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Delete

Deklarace:
procedure Delete(var S: String; Index, Legth: LongInt);

Parametry:
var S: String
Zdrojový øetìzec
Index: LongInt
Index prvního znaku (první v øetìzci je index 1)
Length: LongInt
Poèet znakù, které se mají odstranit.

Popis:
Funkce *Delete* odstraòuje *Length* znakù z øetìzce *S*, index prvního znaku je urèen parametrem *Index*. Délka výsledného øetìzce se bude rovnat hodnotì výrazu "*Length(S) - Length".*

Návratový výraz:
Nevrací žádný výraz pøímo, mìní hodnotu parametru *S*.

Knihovna funkcí - prohlížení

Kategorie: *Utility*
Téma: GetArrayLength

Deklarace:
function GetArrayLength(var Arr: Array): LongInt;

Parametry:
var Arr: Array
Pole, jehož velikost se má zjistit.

Popis:
Funkce *GetArrayLength* zjiš□uje délku dynamického *Arr*. Prvky pole lze pak indexovat podle klíèe "*0* <= *index* < *GetArrayLength*". Délku pole lze mìnit obdobnou funkcí SetArrayLength.

Návratový výraz:
Vrací vždy délku pole, zadaného parametrem *Arr*.

Knihovna funkcí - prohlížení

Kategorie: *Obecné*
Téma: GetVersionEx

Deklarace:
procedure GetVersionEx(var Major, Minor, Release: LongInt)

Parametry:
var Major: LongInt
var Minor: LongInt
var Release: LongInt
Do tìchto parametrù jsou vráceny informace o verzi programu Programmer's Wizard 2.

Popis:
Funkce *GetVersionEx* se používá ke zjištìní informací o verzi programu Programmer's Wizard, na kterém je právì skript spuštìn. Parametr *Major* je v této verzi nastaven na 2, *Minor* na 0 a *Release* na 1. Pøíklad: Programmer's Wizard 3 (v3.1.2) bude vracet hodnoty 3, 1 a 2.

Návratový výraz:
Funkce vždy kopíruje informace o verzi do parametrù *Major*, *Minor* a *Release*.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Halt

Deklarace:
procedure Halt;

Popis:
Funkce *Halt* provede okamžité ukonèení skriptu a pøedá øízení zpìt programu Programmer's Wizard 2.

Návratový výraz:
Nevrací žádný výraz.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Chr

Deklarace:
function Chr(B: Byte): Char

Parametry:
B: Byte
Byte, který se má pøevést na znak

Popis:
Funkce *Chr* pøevádí byte na znak (typ Char). Konverze je užiteèná zvláštì pro vkládání nìkterých speciální znakù do øetìzce, konec øádky je napøíklad možné vytvoøit kombinací *Chr(13)+Chr(10).* Opaèná konverze je možná obdobnou funkcí Ord.

Návratový výraz:
Funkce vždy výsledek pøevedení byte *X* na ASCII znak.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Inc

Deklarace:
procedure Inc(var X: LongInt);

Parametry:
var X: LongInt
Promìnná, jejíž obsah má být inkrementován.

Popis:
Funkce *Dec* zvyšuje hodnotu promìnné *X* o jednu a je ekvivalentní zápisu "*X := X + 1".*

Návratový výraz:
Nevrací žádný výraz pøímo, mìní hodnotu parametru *X*.

Knihovna funkcí - prohlížení

Kategorie: *Utility*
Téma: Insert

Deklarace:
procedure Insert(SubStr: string; var S: String; Index: LongInt)

Parametry:
SubStr S: String
Øetìzec, který má být vložen
var S: String
Cílový øetìzec
Index: LongInt
Pozice, na kterou se má øetìzec *SubStr* vložit

Popis:
Funkce *Insert* vkládá do øetìzce *S* øetìzec *SubStr* a to na pozici udanou parametrem *Index*. Výsledek operace je zkopírován zpìt do parametru *S*. Nová délka øetìzce po této operaci bude "*Length(S) + Length(SubStr)*".

Návratový výraz:
Nevrací žádný výraz pøímo, mìní hodnotu parametru *S*.

Knihovna funkcí - prohlížení

Kategorie: *Utility*
Téma: IntToStr

Deklarace:
function IntToStr(X: LongInt): String

Parametry:
X: LongInt
Ordinální výraz, který má být pøeveden na øetìzec.

Popis:
Funkce *IntToStr* pøevádí parametr *X*, který je ordinální, na øetìzec znakù. Pøíklad: výsledkem volání *IntToStr(123)* bude øetìzec *"123"*. Opaèná konverze je možná voláním funkce StrToInt.

Návratový výraz:
Funkce vždy vrací øetìzec, který vyjadøuje hodnotu parametru *X*.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Length

Deklarace:
function Length(S: String): LongInt

Parametry:
S: String
Øetìzec, jehož délka se má zjistit

Popis:
Funkce *Length* zjiš uje aktuální délku dynamického øetìzce. Øetìzec pak lze indexovat podle klíèe "*1 <= index <= Length(S)*".

Návratový výraz:
Funkce vrací vždy délku øetìzce *S*.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: LowerCase

Deklarace:
function LowerCase(S: String): String

Parametry:
S: String
Øetìzec, jehož obsah má být pøeveden na malá písmena

Popis:
Funkce *LowerCase* pøevádí øetìzec udaný parametrem *S* tak, aby obsahoval jen malá písmena (pø. øetìzec "AbCd" bude pøeveden na "abcd").

Návratový výraz:
Funkce vrací vždy øetìzec *S* pøevedený tak, aby obsahoval jen malé znaky.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Ord

Deklarace:
function Ord(Ch: Char): Byte

Parametry:
Ch: Char
Znak, který se má pøevést na byte

Popis:
Funkce *Ord* pøevádí ASCII znak (typ Char) na typ byte. Konverze je užiteèná zvláštì pro zjiš□ování nìkterých speciální znakù v øetìzci. Opaèná konverze je možná obdobnou funkcí *Chr*.

Návratový výraz:
Funkce vždy výsledek pøevedení ASCII znaku *Ch* na byte.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Pi

Deklarace:
function Pi: Extended

Návratový výraz:
Vždy vrací *3.1415926535897932385*.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Pos

Deklarace:
function Pos(SubStr, S: String): LongInt

Parametry:
SubStr: String
Hledaný øetìzec
S: String
Øetìzec, který se má prohledat.

Popis:
Funkce *Pos* hledá v øetìzci *S* první výskyt øetìzce *SubStr* a po jeho nalezení vrací index prvního znaku *SubStr* v rámci øetìzce *S* (pø. *"Pos('BCD', 'ABCDE)"* vrátí hodnotu "*2*").

Návratový výraz:
Funkce vrací index prvního znaku øetìzce *SubStr* nalezeného v øetìzci *S*. Pokud øetìzec *S* øetìzec *SubStr* neobsahuje, funkce vrací hodnotu nula (*0*).

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Randomize

Deklarace:
procedure Randomize

Popis:
Funkce *Randomize* inicializuje generátor pseudonáhodných èísel. Tato funkce by mìla být zavolána na zaèátku skriptu, který používá funkci *Random* ale nesmí být používána mezi jednotlivými voláními funkce *Random* ve smyèce.

Návratový výraz:
Nevrací žádný výraz.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Random

Deklarace:
function Random(Range: LongInt): LongInt;

Parametry:
Range: LongInt
Rozsah výsledu

Popis:
Funkce *Random* vrací pseudonáhodné èíslo v rozsahu *0* <= *výsledek* < *Range*. Pøed použitím této funkce by mìl být generátor pseudonáhodných èísel nastaven voláním funkce *Randomize*.

Návratový výraz:
Vždy vrací pseudonáhodné èíslo v rozsahu *0* <= *výsledek* < *Range*.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Round

Deklarace:
function Round(X: Extended): LongInt

Parametry:
X: Extended
Výraz, který má být zaokrouhlen

Popis:
Funkce *Round* zaokrouhluje hodnotu udanou výrazem *X* na nebližší hodnotu bez desetinných míst a vrací výsledek jako *LongInt*. Pro pøípady kde je zaokrouhlování nežádoucí je vhodnìjší funkce Trunc.

Návratový výraz:
Funkce vždy vrací hodnotu výrazu *X* zaokrouhlenou na hodnotu bez desetinných míst.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: SetArrayLength

Deklarace:
procedure SetArrayLength(var Arr: Array; NewLength: LongInt)

Parametry:
var Arr: Array
Pole, jehož velikost se má zjistit.

Popis:
Funkce *SetArrayLength* nastavuje délku dynamického pole udaného parametru *Arr* na hodnotu parametru *NewLength*. Po provedení této funkce lze pole indexovat podle tohoto klíèe: "*0* <= *index* < *GetArrayLength(Arr)".* Zjistit velikost dynamického pole lze obdobnou funkcí GetArrayLength.

Návratový výraz:
Nevrací žádný výraz.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Sin

Deklarace:
function Sin(X: Extended): Extended

Parametry:
X: Extended
Parametr funkce Sin

Popis:
Funkce provádí standardní operaci *Sin*, parametrem je *X*. Parametr i výsledek jsou typu *Extended* a jsou chápány v radiánech.

Návratový výraz:
Funkce vždy vrací výsledek operace *Sin X*.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: StrGet

Deklarace:
function StrGet(S: String; Index: LongInt): Char;

Parametry:
S: String
Cílový øetìzec
Index: LongInt
Index požadovaného znaku

Popis:
Funkce *StrGet* vrací *Index*-tý znak z øetìzce *S*. *Index* musí být v rozsahu "*1* <= *S* <= *Length(S)*". Více informací lze nalézt u popisu specifik datového typu String v Pascal Scriptu. Nastavit individuální znak øetìzce lze obdobnou funkcí *StrSet*.

Návratový výraz:
Vrací *Index*-tý znak z øetìzce *S*.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: StrSet

Deklarace:
procedure StrSet(Ch: Char; Index: LongInt; var S: String)

Parametry:
Ch: Char
Znak, na kterou má být prvek øetìzce *S* nastaven
Index: LongInt
Index požadovaného znaku
var S: String
Cílový øetìzec

Popis:
Funkce *StrSet* nastavuje *Index*-tý znak z øetìzce *S* na znak *Ch*. *Index* musí být v rozsahu "*1* <= *S* <= *Length(S)*". Více informací lze nalézt u popisu specifik datového typu String v Pascal Scriptu. Opaènou funkcí je *StrGet*.

Návratový výraz:
Nevrací žádnou hodnotu pøímo, mìní øetìzec udaný parametrem *S*.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: StrToInt

Deklarace:
function StrToInt(S: String; Default: LongInt): LongInt

Parametry:
S: String
Øetìzec, který má být pøeveden na ordinální výraz
Default: LongInt
Ordinální výraz, který funkce vrátí, když nelze øetìzec pøevést

Popis:
Funkce *StrToInt* pøevádí øetìzec vyjadøující ordinální hodnotu (jako je øetìzec "*123*") na ordinální datový typ. Opaèná konverze je možná voláním funkce *IntToStr*.

Návratový výraz:
Funkce vrací ordinální hodnotu, vyjádøenou øetìzcem *S*. Pokud øetìzec nelze pøevést (tj. pokud obsahuje jiné znaky než èíslice), vrací funkce hodnotu udanou parametrem *Default*.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: Round

Deklarace:
function Trunc(X: Extended): LongInt

Parametry:
X: Extended
Výraz, který má být zbaven desetinných míst

Popis:
Funkce *Trunc* pøevádí výraz *X* na typ *LongInt* bez zaokrouhlování, prostým odstranìním desetinné èásti výrazu. Pro situace, kde je zapotøebí zaokrouhlování pøed pøevodem je vhodnìjší funkce Round.

Návratový výraz:
Funkce vždy vrací hodnotu výrazu *X* bez desetinných míst.

Knihovna funkcí - prohlížení

Kategorie: Utility
Téma: UpperCase

Deklarace:
function UpperCase(S: String): String

Parametry:
S: String
Øetìzec, jehož obsah má být pøeveden na velká písmena

Popis:
Funkce *UpperCase* pøevádí øetìzec udaný parametrem *S* tak, aby obsahoval jen velká písmena (pø. øetìzec "AbCd" bude pøeveden na "ABCD").

Návratový výraz:
Funkce vrací vždy øetìzec *S* pøevedený tak, aby obsahoval jen velké znaky.

Knihovna funkcí - prohlížení

Kategorie: *Registr*
Téma: RegReadInt

Deklarace:
function RegReadInt(Name: String; DefaultValue: LongInt): LongInt

Parametry:
Name: String
Øetìzec udávající název hodnoty, která se má pøeèíst
DefaultValue: LongInt
Hodnota, která má být vrácena v pøípadì, že hodnota s názvem *Name* neexistuje.

Popis:
Funkce *RegReadInt* ète z registru døíve uloženou pojmenovanou hodnotu. Takovou hodnotu lze zapsat voláním funkce *RegWriteInt*, hodnota pak zùstává v registru uložena pro pozdìjší použití pøi pøíštím spuštìní skriptu. Všechny skripty sdílejí stejnou složku v registru.

Návratový výraz:
Funkce ète hodnotu pojmenovanou *Name* a vrací ji jako typ *LongInt*. Pokud hodnota neexistuje, vrací hodnotu *DefaultValue*.

Poznámky:
Výchozí umístìní složky registru, která se používá pro tyto operace:
HKEY_CURRENT_USER\Software\Programmers Wizard\Scripting

Knihovna funkcí - prohlížení

Kategorie: *Registr*
Téma: RegReadString

Deklarace:
function RegReadString(Name: String; DefaultValue: String): String

Parametry:
Name: String
Øetìzec udávající název hodnoty, která se má pøeèíst
DefaultValue: String
Hodnota, která má být vrácena v pøípadì, že hodnota s názvem *Name* neexistuje.

Popis:
Funkce *RegReadString* ète z registru døíve uloženou pojmenovanou hodnotu. Takovou hodnotu lze zapsat voláním funkce *RegWriteString*, hodnota pak zùstává v registru uložena pro pozdìjší použití pøi pøíštím spuštìní skriptu. Všechny skripty sdílejí stejnou složku v registru.

Návratový výraz:
Funkce ète hodnotu pojmenovanou *Name* a vrací ji jako typ String. Pokud hodnota neexistuje, vrací hodnotu *DefaultValue*.

Poznámky:
Výchozí umístìní složky registru, která se používá pro tyto operace:
HKEY_CURRENT_USER\Software\Programmers Wizard\Scripting

Knihovna funkcí - prohlížení

Kategorie: *Registr*
Téma: RegWriteInt

Deklarace:
function RegWriteInt(Name: String; Value: LongInt): Boolean

Parametry:
Name: String
Øetìzec udávající název hodnoty, která se má zapsat
Value: LongInt
Vlastní hodnota

Popis:
Funkce *RegWriteInt* se používá k uložení libovolných informací do registru. V registru jsou uložené hodnoty organizovány jako dvojice *NÁZEV = HODNOTA*. Této dvojici odpovídají parametry *Name* a *Value*. Skript mùže použít tuto funkcí pro zapsání informací, které bude potøebovat pøi dalším spuštìní, protože tyto informace v registru zùstávají i po ukonèení skriptu. Pøi dalším spuštìní mùže skript použít funkce *RegReadInt* a svou hodnotu si pøeèíst.

Návratový výraz:
Funkce vrací *TRUE* pokud vše probìhlo bez problémù a hodnota byla zapsána, jinak vrací *FALSE*.

Poznámky:
Výchozí umístìní složky registru, která se používá pro tyto operace:
HKEY_CURRENT_USER\Software\Programmers Wizard\Scripting

Knihovna funkcí - prohlížení

Kategorie: Registr
Téma: RegWriteString

Deklarace:
function RegWriteString(Name: String; Value: LongInt): Boolean

Parametry:
Name: String
Øetìzec udávající název hodnoty, která se má zapsat
Value: LongInt
Vlastní hodnota

Popis:
Funkce *RegWriteString* se používá k uložení libovolných informací do registru. V registru jsou uložené hodnoty organizovány jako dvojice *NÁZEV* = *HODNOTA*. Této dvojici odpovídají parametry *Name* a *Value*. Skript mùže použít tuto funkcí pro zapsání informací, které bude potøebovat pøi dalším spuštìní, protože tyto informace v registru zùstávají i po ukonèení skriptu. Pøi dalším spuštìní mùže skript použít funkce RegReadString a svou hodnotu si pøeèíst.

Návratový výraz:
Funkce vrací *TRUE* pokud vše probìhlo bez problémù a hodnota byla zapsána, jinak vrací *FALSE*.

Poznámky:
Výchozí umístìní složky registru, která se používá pro tyto operace:
HKEY_CURRENT_USER\Software\Programmers Wizard\Scripting

Registrace

Po úspìšné instalaci programu je potøeba Vaši kopii zaregistrovat. Registrace je bezplatná, zabere Vám jen pár minut a odesílá se pomocí e-mailu na adresu pwiz@volny.cz nabo petr.esner@atlas.cz. Vìnujte prosím tìch nìkolik málo minut vyplnìní registraèního prùvodce, pomùže mi to v udržení statistik o využívanosti programu a poètu uživatelù. Pokud nevidím ohlas uživatelù, jaký budu mít dùvod k pokraèování vývoje tohoto programu?
Program samotný nebude nijak urgovat registraci, nikdy nezaène zobrazovat zdržující dialogy s upozornìním na nutnou registraci, nebude zobrazovat žádnou reklamu a nikdy neomezí nìkteré své funkce, aby tak donutil uživatele k registraci. Pøesto (nebo právì proto, oceníte takový pøístup?) je potøeba program zaregistrovat.
Registraci mùžete spustit ze stránek pwiz.hyperlink.cz.
Registrací nezískáte jen dobrý pocit :-), získáte hlavnì pøednostní pøístup k technické podpoøe a jistotu, že Vaše e-maily nikdy nebudou odbyty automatickou odpovìdí (tímto se omlouvám neregistrovaným uživatelùm; je velmi nároèné odpovídat velkému množství uživatelù na stejné dotazy a reagovat na podobné pøipomínky).
Poznámka pro vlastníky verze Programmer's Wizard 2000:
I když je Programmer's Wizard 2000 zaregistrován, je potøeba zaregistrovat verzi Programmer's Wizard 2 zvláš. Registrace pak samozøejmì není nutná u dalších verzí programu Programmer's Wizard 2 pøípadnì pøi stažení aktualizovaných zásuvných modulù.

Autor

Knihovna funkcí - prohlížení

Kategorie: Obecné
Téma: SendMail

Deklarace:
function SendMail(To, Subject, Text: String): Boolean

Parametry:
To: String
E-mailová adresa pøíjemce
Subject: String;
Titulek zprávy
Text: String
Vlastní text zprávy

Popis:
Funkce *SendMail* odesílá emailovou zprávu. Parametr *To* udává adresu nebo adresy pøíjemce(ù), jednotlivé adresy musí být oddìleny støedníkem. *Subject* obsahuje titulek (subjekt) zprávy a v parametru *Text* by mìl být vlastní text zprávy. Funkce pøi svém zavolání také zobrazí dialog, ve kterém mùže uživatel pozmìnit text zprávy, pøípadnì pøidat pøílohy.
Tato funkce neumožòuje nastavení pøíjemcù kopie zprávy ani pøikládání pøíloh, pro tyto dodateèné funkce lze použít funkci *SendMailEx*.

Návratový výraz:
Funkce vrací *TRUE* pokud byla zpráva úspìšnì odeslána, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: Obecné
Téma: SendMail

Deklarace:
function SendMailEx(var ToList, CcList, AttachmentList: Array of String; Subject, Text: String): Boolean

Parametry:
var ToList: Array of String
Pole se seznamem pøíjemcù zprávy
var CcList: Array of String
Pole se seznamem pøíjemcù kopie zprávy
var AttachmentList: Array of String
Pole se seznamem souborù, které mají být ke zprávì pøiloženy
Subject: String;
Titulek zprávy
Text: String
Vlastní text zprávy

Popis:
Funkce *SendMailEx* odesílá emailovou zprávu. Parametr *ToList* je dynamické pole a jeho prcky udávají adresy pøíjemce(ù). *CcList* obsahuje stejným zpùsobem uložené adresy pøíjemcù kopií zprávy a pole *AttachmentList* mùže obsahovat seznam souborù, které mají být pøiloženy. *Subject* obsahuje titulek (subjekt) zprávy a v parametru *Text* by mìl být vlastní text zprávy. Funkce pøi svém zavolání také zobrazí dialog, ve kterém mùže uživatel pozmìnit text zprávy, pøípadnì pøidat pøílohy.
Pro pøípady kde není potøeba parametrù *CcList* a *AttachmentList* je vhodnìjší funkce SendMail .

Návratový výraz:
Funkce vrací *TRUE* pokud byla zpráva úspìšnì odeslána, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: Obecné
Téma: EnumFiles

Deklarace:
procedure EnumFiles(var Files: Array of String)

Parametry:
var Files: Array of String
Do tohoto parametru je vrácen seznam otevøených dokumentù

Popis:
Funkce *EnumFiles* se používá k získání informací o právì otevøených dokumentech. Funkce vrací seznam dokumentù do parametru *Files*, seznamem je myšlen výpis všech pojmenuvaných dokumentù, které jsou právì na otevøené ploše; nezáleží na tom, zda je otevøen projekt. Prvky pole *Files* budou obsahovat plné cesty k otevøeným dokumentùm, skript pak mùže zjistit velikost pole a tím i poèet dokumentù volání funkce GetArrayLength.

Návratový výraz:
Funkce nevrací žádnou hodnotu pøímo, mìní parametr *Files*.

Poznámky:
Tato funkce vytvoøí seznam otevøených dokumentù, ignoruje nové, ještì nepojmenované dokumenty. Pokud potøebuje skript i tuto informaci, mìl by použít funkci *EnumWindows*.

Knihovna funkcí - prohlížení

Kategorie: *Okna*
Téma: EnumWindows

Deklarace:
procedure EnumWindows(var Handles: Array of LongInt)

Parametry:
var Handles: Array of LongInt
Do tohoto parametru je vrácen seznam otevøených oken programu

Popis:
Funkci *EnumWindows* lze použít pro zjištìní stavu plochy programu. Funkce vytváøí seznam všech otevøených oken a vkládá jej do parametru *Handles*. Seznamem oken jsou zde myšlena všechna okna nástrojù vèetnì oken nástrojù pro editaci dokumentù (jako je napø. SyntaxEditor) i oken jiných nástrojù, napø. nástroje Process Navigator. Vrácený seznam obsahuje identifijátory oken, které lze použít v dalších funkcích a tím zjistit více podrobností.

Návratový výraz:
Funkce nevrací žádnou hodnotu pøímo, mìní parametr *Handles*.

Poznámky:
Tato funkce vytvoøí seznam všech otevøených oken vèetnì nových, ještì nepojmenovaných dokumentù. Pokud skript nepotøebuje informace o nepojmenovaných dokumentech, mìl by použít funkci *EnumFiles*.

Knihovna funkcí a procedur (Pascal Script)
(zpìt do obsahu knihovny tudy)

Termíny použité v této kategorii
Kategorie Okna
EnumWindows
GetCurrentWindow
IsWindow
WindowToFileName
WindowToPluginID
WindowToPluginEntryID

Odkazy:
Skriptování
Datové typy
Cykly a pøíkazy
Matematické operace
Pøíklady

Knihovna funkcí - prohlížení

Kategorie: Okna
Téma: WindowToFileName

Deklarace:
function WindowToFileName(Handle: LongInt; var FileName: String): Boolean

Parametry:
Handle: LongInt
Identifikátor okna
var FileName: String
Do tohoto parametru je vrácen název dokumentu, upravovaném v oknì *Handle*.

Popis:
Funkce *WindowToFileName* pøevádí identifikátor (*Handle*) okna na název dokumentu, který se v daném oknì právì upravuje. Název dokumentu funkce zkopíruje do parametru *FileName*.

Návratový výraz:
Funkce vrací *TRUE* pokud vše probìhlo bez chyb a název dokumentu byl zkopírován do parametru *FileName*. Pokud okno neexistuje, vrací funkce *FALSE*.

Poznámky:
Pro získání seznamu všech existujících oken by skript mìl použít funkci EnumWindows.

Knihovna funkcí - prohlížení

Kategorie: Okna
Téma: IsWindow

Deklarace:
function IsWindow(Handle: LongInt): Boolean

Parametry:
Handle: LongInt
Identifikátor okna

Popis:
Funkce *IsWindow* se používá ke zjištìní, zda okno s identifikátorem *Handle* existuje. Oknem je zde myšleno okno na ploše programu Programmer's Wizard 2, vìtšinou okno nìkterého nástroje.

Návratový výraz:
Funkce vrací *TRUE* pokud okno existuje, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: Okna
Téma: WindowToPluginID

Deklarace:
function WindowToPluginID(Handle: LongInt; var PluginID: String): Boolean

Parametry:
Handle: LongInt
Identifikátor okna
var PluginID: String
Do tohoto parametru je vrácen název dokumentu, upravovaném v oknì *Handle*.

Popis:
Funkce *WindowToFileName* pøevádí identifikátor (*Handle*) okna na název dokumentu, který se v daném oknì právì upravuje. Název dokumentu funkce zkopíruje do parametru *FileName*.

Návratový výraz:
Funkce vrací *TRUE* pokud vše probìhlo bez chyb a název dokumentu byl zkopírován do parametru *FileName*. Pokud okno neexistuje, vrací funkce *FALSE*.

Poznámky:
Pro získání seznamu všech existujících oken by skript mìl použít funkci EnumWindows.

Knihovna funkcí - prohlížení

Kategorie: Okna
Téma: WindowToPluginEntryID

Deklarace:
function WindowToPluginEntryID(Handle: LongInt; var PluginEntryID: String): Boolean

Parametry:
Handle: LongInt
Identifikátor okna
var PluginEntryID: String
Do tohoto parametru je vrácen název dokumentu, upravovaném v oknì *Handle*.

Popis:
Funkce *WindowToFileName* pøevádí identifikátor (*Handle*) okna na název dokumentu, který se v daném oknì právì upravuje. Název dokumentu funkce zkopíruje do parametru *FileName*.

Návratový výraz:
Funkce vrací *TRUE* pokud vše probìhlo bez chyb a název dokumentu byl zkopírován do parametru *FileName*. Pokud okno neexistuje, vrací funkce *FALSE*.

Poznámky:
Pro získání seznamu všech existujících oken by skript mìl použít funkci EnumWindows.

Knihovna funkcí - prohlížení

Kategorie: *Okna*
Téma: Použité termíny

"Identifikátor okna **Handle**"
V této kategorii se èasto používá a mluví o identifikátoru okna *Handle*. Identifikátorem *Handle* je zde myšleno suteèné *Handle* systému Windows (více informací o tomto datovém typu v nápovìdì pro WinAPI). Nìkteré funkce této kategorie mají shodná jména jako funkce Windows, jde o funkce jako *EnumWindows* a *IsWindow*. Tyto funkce pracují obdobnì jako jejich WinAPI ekvivalenty s tím rozdílem, že pracují pouze v kontextu plochy programu Programmer's Wizard 2.

"Okna na ploše programu"
Pro aplikace systému Windows, kontextem pro funkce typu EnumWindows je samotný OS Windows a funkcí lze zjistit výpis všech dceøinných oken k danému rodièovskému. Pro skript programu Programmer's Wizard 2 je kontextem a jediným rodièovským oknem MDI klient aplikace, tedy plocha, která obsahuje všechny okna nástrojù.

Knihovna funkcí - prohlížení

Kategorie: *Projekt*
Téma: Použité termíny

"identifikátor projektu **ProjectID**"
V programu Programmer's Wizard 2 jsou všechny projekty uspoøádány do pole prvkù, které lze adresovat jako pole Pascal Scriptu. Prvním prvkem pole (a tedy i prvním projektem) je projekt s indexem 0, další jsou 1, 2 atd. Index posledního prvku v poli lze zjistit voláním funkce *GetProjectCount*, platné hodnoty ProjectID jsou pak v rozsahu *"0 <= index < GetProjectCount"*. Skript nemùže pøedpokládat, že toto pole prvkù je nemìnné; pole se mùže mìnit za bìhu programu tak, jak uživatel vytváøí nové projekty; mùže být také zmìnìno samotným skriptem za jeho bìhu funkcemi z této kategorie.

Knihovna funkcí - prohlížení

Kategorie Plochy
Téma: Použité termíny

"identifikátor projektu **WorkspaceID**"
V programu Programmer's Wizard 2 jsou všechny plochy projektu uspoøádány do pole prvkù, které lze adresovat jako pole Pascal Scriptu. Prvním plochou pole je plocha s indexem 0, další jsou 1, 2 atd. Index posledního prvku v poli lze zjistit voláním funkce *GetWorkspaceCount*, resp. *GetWorkspaceCountEx*. Platné hodnoty WorkspaceID jsou pak v rozsahu *"0 <= index < GetWorkspaceCount/GetWorkspaceCountEx"*. Skript nemùže pøedpokládat, že toto pole prvkù je pro každý projekt nemìnné; pole se mùže mìnit za bìhu programu tak, jak uživatel vytváøí nové plochy a odebírá existující; mùže být také zmìnìno samotným skriptem za jeho bìhu funkcemi z této kategorie.

Knihovna funkcí a procedur (Pascal Script)
(zpìt do obsahu knihovny tudy)

Termíny použité v této kategorii
Kategorie Zásuvné moduly
EnumPluginIDs
EnumPluginEntryIDs
PluginEntryIDToName
PluginIDToFullName
PluginIDToShortName

Odkazy:
Skriptování
Datové typy
Cykly a pøíkazy
Matematické operace
Pøíklady

Knihovna funkcí - prohlížení

Kategorie Zásuvné moduly
Téma: Použité termíny

"identifikátor **PluginID**"
V programu Programmer's Wizard 2 má každý zásuvný modul svùj identifikátor, zvaný *PluginID*. Toto ID je øetìzec, který se nemìní pøi pøidávání a odebírání modulù. Programmer's Wizard 2 po svém naètení vždy vytvoøí seznam svých modulù a pøiøadí jim jejich *PluginID*. Skript mùže za všech okolností pøedpokládat, že žádné dva zásuvné moduly nedostanou pøiøazeno stejné *PluginID*.
PluginID je øetìzec ve formátu: **_pluginXXXXXXXX**, kde *XXXXXXXX* je osmimístný hexadecimální výraz, vypoèítaný programem a pøiøazený modulu.

"identifikátor **PluginEntryID**"
V programu Programmer's Wizard 2 mùže každý zásuvný modul obsahovat jeden nebo více nástrojù. Každý z tìchto nástrojù má svùj identifikátor, zvaný *PluginEntryID*. Identifikátor obsahuje dvì èásti: *PluginID* a index nástroje v rámci modulu. *PluginEntryID* je unikátním øetìzcem pro každý nástroj, který je naèten. Stejnì jako u identifikátoru *PluginID*, i zde mùže skript poèítat s nemìnností výrazu.
*PluginEntryID je øetìzec ve formátu: PluginID**_entryX***, kde *PluginID* je identifikátor modulu a *X* je index nástroje.

Knihovna funkcí - prohlížení

Kategorie: Zásuvné moduly
Téma: PluginIDToFullName

Deklarace:
function PluginIDToFullName(PluginID: String; var FullName: String): Boolean

Parametry:
PluginID: String
Identifikátor modulu
var FullName: String
Do tohoto parametru je vrácen plný název modulu, definovaného parametrem *PluginEntryID*

Popis:
Funkce *PluginIDToFullName* pøevádí identifikátor (*PluginID*) modulu na jeho plný název. Tento název funkce zkopíruje do parametru *FullName*. Pro získání krátkého jména modulu mùže skript použít funkci PluginIDToShortName.

Návratový výraz:
Funkce vrací *TRUE* pokud vše probìhlo bez chyb a název modulu byl zkopírován do parametru *FullName*. Pokud modul definovaný parametrem *PluginID* neexistuje, vrací funkce *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: *Zásuvné moduly*
Téma: PluginIDToShortName

Deklarace:
function PluginIDToShortName(PluginID: String; var ShortName: String): Boolean

Parametry:
PluginID: String
Identifikátor modulu
var ShortName: String
Do tohoto parametru je vrácen krátký název modulu, definovaného parametrem *PluginEntryID*

Popis:
Funkce *PluginIDToShortName* pøevádí identifikátor (*PluginID*) modulu na jeho krátký název. Tento název funkce zkopíruje do parametru Short*Name*. Pro získání plného jména modulu mùže skript použít funkci PluginIDToFullName.

Návratový výraz:
Funkce vrací *TRUE* pokud vše probìhlo bez chyb a název modulu byl zkopírován do parametru *ShortName*. Pokud modul definovaný parametrem *PluginID* neexistuje, vrací funkce *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: Zásuvné moduly
Téma: PluginEntryIDToName

Deklarace:
function PluginEntryIDToName(PluginEntryID: String; var Name: String): Boolean

Parametry:
PluginEntryID: String
Identifikátor modulu
var Name: String
Do tohoto parametru je vrácen název nástroje, definovaného parametrem *PluginEntryID*

Popis:
Funkce *PluginEntryIDToName* pøevádí identifikátor (*PluginEntryID*) nástroje modulu na jeho název. Tento název funkce zkopíruje do parametru *Name*.

Návratový výraz:
Funkce vrací *TRUE* pokud vše probìhlo bez chyb a název nástroje modulu byl zkopírován do parametru *Name*. Pokud nástroj definovaný parametrem *PluginEntryID* neexistuje, vrací funkce *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: *Zásuvné moduly*
Téma: EnumPluginIDs

Deklarace:
procedure EnumPluginIDs(var PluginIDs: Array of String)

Parametry:
var PluginIDs: Array of String
Pole, do kterého funkce zkopíruje seznam nanistalovaných modulù

Popis:
Funkce *EnumPluginIDs* vytváøí seznam všech nainstalovaných zásuvných modulù a kopíruje jej do parametru *PluginIDs*. Více informací o modulech lze získat použitím dalších funkcí této kategorie. Skript mùže procházet nástroje nalezených modulù funkcí EnumPluginEntryIDs.

Návratový výraz:
Nevrací žádnou hodnotu pøímo, mìní parametr *PluginIDs*.

Knihovna funkcí - prohlížení

Kategorie: Zásuvné moduly
Téma: EnumPluginEntryIDs

Deklarace:
function EnumPluginEntryIDs(PluginID: String; var PluginEntryIDs: Array of String): Boolean

Parametry:
PluginID: String
Identifikátor modulu, jehož nástroje má funkce zjistit
var PluginEntryIDs: Array of String
Pole, do kterého funkce zkopíruje seznam nástrojù daného modulu

Popis:
Funkce *EnumPluginEntryIDs* vytváøí seznam všech nástrojù, které jsou dostupné v modulu definovaném parametrem *PluginID*. Funkce kopíruje vytvoøený seznam do parametru *PluginEntryIDs*. Pro zjištìní nainstalovaných modulù a jejich identifikátorù *PluginID* mùže skript použít funkce EnumPluginIDs.

Návratový výraz:
Funkce vrací *TRUE* pokud vše probìhlo v poøádku a seznam byl úspìšnì vytvoøen a zkopírován. Pokud modul *PluginID* neexistuje, vrací funkce *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: *Obecné*
Téma: GetCurrentScript

Deklarace:
function GetCurrentScript: String

Popis:
Funkce *GetCurrentScript* vrací název a plnou cestu k souboru, ze kterého byl právì bìžící skript naèten.

Návratový výraz:
Funkce vrací vždy název a plnou cestu k souboru, ze kterého byl skript naèten.

Poznámky:
Skript by mìl tuto funkci používat v souvislosti s funkcemi registru *RegReadInt*, *RegReadString*, *RegWriteInt* a *RegWriteString*. Pøed voláním tìchto funkcí by skript vždy mìl použít název sebe sama jako prefix k názvùm hodnot, které chce zapisovat a èíst, tak lze zabránit pøípadným konfliktùm.

Knihovna funkcí - prohlížení

Kategorie: *Okna*
Téma: GetCurrentWindow

Deklarace:
function GetCurrentWindow: LongInt

Popis:
Funkce *GetCurrentWindow* vrací *Handle* okna, které je právì aktivní. Pokud je otevøeno jen jedno okno, funkce použije jeho handle, pøi více oknech vrací funkce handle toho okna, které je právì aktivní.

Návratový výraz:
Funkce vrací *Handle* právì aktivního okna. Pokud není otevøeno žádné okno, vrací funkce hodnotu nula (0).

Knihovna funkcí - prohlížení

Kategorie: Obecné
Téma: Sleep

Deklarace:
procedure Sleep(Milliseconds: LongInt);

Parametry:
Milliseconds: LongInt
Interval, na který má být skript zmrazen

Popis:
Funkce *Sleep* pozastaví provádìní skriptu na dobu uvedenou v parametru *Milliseconds*. I když je skript zmrazen, uživatel mùže stále použít funkci *Pøerušit skript* a tím bìžící skript ukonèit.

Návratový výraz:
Nevrací žádný výraz.

Knihovna funkcí - prohlížení

Kategorie: *Obecné*
Téma: OpenFileEx

Deklarace:
function OpenFileEx(FileName, PluginEntryID: String): Boolean

Parametry:
FileName: String
Název dokumentu k otevøení
PluginEntryID: String
Identifikátor nástroje, kterým se má modul otevøít

Popis:
Funkce *OpenFileEx* otevírá dokument udaný parametrem *FileName* v programu Programmer's Wizard 2. Dokument se otevøe nástrojem, který je identifikován parametrem *PluginEntryID* (více informací zde). Tato funkce vždy otevøe soubor v programu Programmer's Wizard 2, pro spuštìní programu nebo otevøení dokumentu jeho asociovaným program je vhodnìjší funkce *ExecuteFile*.

Návratový výraz:
Funkce vrací *TRUE* pokud byl soubor úspìšnì otevøen, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: Pøímá komunikace s uživatelem
Téma: ShowCheckListDialog

Deklarace:
function ShowCheckListDialog(Caption, Prompt: String; var Lines: Array of String; var CheckBoxes: Array of boolean): LongInt

Parametry:
Caption: String
Titulek okna
Prompt: String
Text uvnitø okna
var Lines: Array of String
Seznam øetìzcù, které se mají zobrazit jako jednotlivé položky
var CheckBoxes: Array of boolean
Pole popisující stav zaškrtávacích tlaèítek u jednotlivých položek

Popis:
Funkce *ShowCheckListDialog* pracuje obdobnì jako funkce ShowListDialog, navíc dokáže vedle každé položky v seznamu zobrazit zaškrtávací tlaèítko. Uživatel mùže zaškrtnout nebo odškrtnout jakékoliv tlaèítko. Pøi otevírání okna jsou tlaèítka nastavena na hodnoty, udané parametrem *CheckBoxes*; každému prvku z pole *Lines* odpovídá pøíslušný prvek z *CheckBoxes*. Pokud je prvek pole nastaven na hodnotu *TRUE*, bude tlaèítko na pøíslušném øádku zaškrtnuto, pøi hodnotì *FALSE* bude èisté. Uživatel mùže zaškrtáváním a odškrtáváním tlaèítek mìnit stav pole *CheckBoxes*.

Návratový výraz:
Funkce vrací index vybraného øetìzce. Tento index lze pøímo použít do pole *Lines* takto: "Lines[result]" kde *result* je hodnota vrácená funkcí ShowListDIalog. Když uživatel zruší výbìr prvku tlaèítkem *Storno*, funkce vrací hodnotu -1. Skript musí vždy ovìøit, zda není výsledná hodnota -1 protože bez tohoto testu by mohla být provedena tato instrukce: "Lines[-1]". Instrukce je neplatná protože první prvek z pole má index 0 - skript by byl okamžitì ukonèen s chybovým hlášením. Stav zaškrtávacích tlaèítek je vrácen do pole *CheckBoxes* a pøistupovat k nìmu lze stejnì, jako k poli *Lines*.

Knihovna funkcí - prohlížení

Kategorie: FS (soubory a složky)
Téma: CopyFile

Deklarace:
function CopyFile(ExistingName, NewName: String; FailIfExists: Boolean): Boolean

Parametry:
ExistingName: String
Plná cesta k existujícímu souboru
NewName: String
Nový název souboru

Popis:
Funkce *CopyFile* pøejmenuje fyzický soubor udaný parametrem *ExistingName* na název *NewName*.

Návratový výraz:
Funkce vrací *TRUE* pokud byl soubor smazán, jinak vrací *FALSE*.

Poznámky:
Tato funkce je jednou z chránìných funkcí a mùže být v nastavení skriptování zakázána. Skript by si mìl vždy ovìøit zda jsou tyto funkce povoleny voláním *IsSafeScriptingEnabled*.

Knihovna funkcí - prohlížení

Kategorie: *Obecné*
Téma: GetProjectsDir

Deklarace:
function GetProjectsDir: String

Popis:
Funkce *GetProjectsDir* se používá ke zjištìní cesty ke složce, ve které jsou uloženy všechny projekty.

Návratový výraz:
Funkce vrací vždy plnou cestu ke složce, ve které jsou uloženy všechny projekty.

Knihovna funkcí - prohlížení

Kategorie: Obecné
Téma: GetScriptsDir

Deklarace:
function GetScriptsDir: String

Popis:
Funkce *GetScriptsDir* se používá ke zjištìní cesty ke složce, ve které jsou uloženy všechny skripty. Uživatel ale mùže spustit skript z jakékoliv složky a bìžící skript proto nemùže pøedpokládat, že je spuštìn ze složky vrácené touto funkcí. Pro zjíštìní aktuální složky, ze které je skript spuštìn, je vhodnìjší funkce GetCurrentScript.

Návratový výraz:
Funkce vrací vždy plnou cestu ke složce, ve které jsou uloženy všechny skripty.

Knihovna funkcí - prohlížení

Kategorie: Pøímá komunikace s uživatelem
Téma: SelectDirectory

Deklarace:
function SelectDirectory(Caption: String; var Directory: String): Boolean

Parametry:
Caption: String
Titulek okna
var Directory: String
Do tohoto parametru je pøi uzavøení okna zkopírována složka, kterou uživatel vybral

Popis:
Funkce *SelectDirectory* zobrazuje dialogové okno, ve kterém mùže uživatel vybrat libovolný fyzický adresáø. Funkci lze použít všude tam, kde má mít uživatel možnost vybrat složku.

Návratový výraz:
Funkce vrací *TRUE* pokud bylo stisknuto tlaèítko OK a parametr *Directory* nyní obsahuje plnou cestu k vybrané složce, jinak vrací *FALSE*.

Knihovna funkcí - prohlížení

Kategorie: Pøímá komunikace s uživatelem
Téma: SelectFiles

Deklarace:
function SelectFiles(InitialDir, Filter: String; var Files: Array of String): Boolean

Parametry:
InitialDir: String
Titulek okna
Filter: String
Tento parametr udává filtr, který se má zobrazit v dialogu Otevøít. Filtr musí mít tento formát:
'*Název typu souboru 1*|*.*pøípona souboru 1*|' +
'*Název typu souboru 2*|*.*pøípona souboru 2*|' +
'*Název typu souboru 3*|*.*pøípona souboru 3*'
var Files: String
Do tohoto parametru je pøi uzavøení okna zkopírován seznam vybraných souborù

Popis:
Funkce *SelectFiles* dovoluje zobrazit dialog, ve kterém mùže uživatel vybrat libovolný existující soubor nebo soubory. Parametr *InitialDir* urèuje výchozí adresáø; ten bude vybrán po otevøení okna. Parametr *Filter* udává seznam položek v seznamu *Typ souboru*, více informací dále.

Návratový výraz:
Funkce vrací *TRUE* pokud bylo stisknuto tlaèítko OK a parametr *Files* nyní obsahuje seznam všech vybraných souborù, jinak vrací *FALSE*.

Pøíklad parametru *Filter*:
Filter := "Textové soubory (.txt;*.log)|*.txt;*.log|Wordpad (*.rtf)|*.rtf"*

Knihovna funkcí a procedur (Pascal Script)
(zpìt do obsahu knihovny tudy)

Dokumenty (soubory a složky)
EnumDocuments
SetDocuments

Odkazy:
Skriptování
Datové typy
Cykly a pøíkazy
Matematické operace
Pøíklady

Knihovna funkcí - prohlížení

Kategorie: *Dokumenty*
Téma: EnumDocuments

Deklarace:
procedure EnumDocuments(var Documents: Array of String)

Parametry:
var Documents: Array of String
Do tohoto parametru je vrácen seznam souborù ve složce dokumentù

Popis:
Funkci *EnumDocuments* lze použít pro zjištìní stavu virtuální složky Dokumenty. Funkce vrací seznam souborù ve složce Dokumenty do parametru *Documents*.

Návratový výraz:
Funkce nevrací žádnou hodnotu pøímo, mìní parametr Documents.

Knihovna funkcí - prohlížení

Kategorie: *Dokumenty*
Téma: SetDocuments

Deklarace:
procedure SetDocuments(var Documents: Array of String)

Parametry:
var Documents: Array of String
Obsah tohoto parametru je zkopírován do složky Dokumenty

Popis:
Funkce *SetDocuments* mìní obsah složky Dokumenty tak, aby odpovídal seznamu souborù, udaného parametrem *Documents*.

Návratový výraz:
Nevrací žádný výraz.

Knihovna funkcí - prohlížení

Kategorie: Projekt
Téma: EnumProjectFiles

Deklarace:
function EnumProjectFiles(ProjectID: LongInt; var Files: Array of String): Boolean

Parametry:
ProjectID: LongInt
ID projektu
var Files: Array of String
Do tohoto parametru je zkopírován seznam souborù v projektu

Popis:
Funkce *EnumProjectFiles* získává seznam všech souborù, které jsou v projektu definovaném parametrem *ProjectID*. Tento seznam pak kopíruje do parametru *Files*.

Návratový výraz:
Funkce vrací *TRUE* pokud byly informace zkopírovány do parametrù, jinak vrací *FALSE*. Funkce vìtšinou vrátí *FALSE* jako následek nesprávného parametru *ProjectID*.

Knihovna funkcí - prohlížení

Kategorie: Projekt
Téma: SetProjectFiles

Deklarace:
function SetProjectFiles(ProjectID: LongInt; var Files: Array of String): Boolean

Parametry:
ProjectID: LongInt
ID projektu
var Files: Array of String
Obsah tohoto parametru je zkopírován do seznamu souborù v daném projektu

Popis:
Funkce *SetProjectFiles* mìní seznam souborù v projektu *ProjectID* tak, aby odpovídat seznamu souborù definovaným parametrem *Files*.

Návratový výraz:
Funkce vrací *TRUE* pokud byly informace zkopírovány, jinak vrací *FALSE*. Funkce vìtšinou vrátí *FALSE* jako následek nesprávného parametru *ProjectID*.

Knihovna funkcí - prohlížení

Kategorie: *Projekt*
Téma: CreateProject

Deklarace:
function CreateProject(Name, Description: String): LongInt

Parametry:
Name: String
Název nového projektu
Description: String
Popis nového projektu

Popis:
Funkce *CreateProject* vytváøí nový projekt a vrací jeho identifikátor *ProjectID*. Nový projekt bude mít název *Name* a popis *Description*, projektu bude také pøiøazena jeho primární plocha. Skript by nikdy nemìl vytváøet projekt bez jména, tedy pøíkazem "*CreateProject(", ")*". Projekt není po vytvoøení automaticky otevøen, pro otevøení projektu mùže skript použít funkci *OpenProject*.

Návratový výraz:
Funkce vrací vždy *ProjectID* vytvoøeného projektu.

Knihovna funkcí - prohlížení

Kategorie: *Plochy*
Téma: CreateWorkspace

Deklarace:
function CreateWorkspace(Name, Description: String): LongInt

Parametry:
Name: String
Název nové plochy
Description: String
Popis nové plochy

Popis:
Funkce *CreateWorkspace* vytváøí novou plochu právì otevøeného projektu a vrací její identifikátor *WorkspaceID*. Nová plocha bude bude mít název *Name* a popis *Description.* Skript by nikdy nemìl vytváøet plochu bez jména, tedy pøíkazem "*CreateWorkspace(", ")*". Plocha není po vytvoøení automaticky otevøena, pro otevøení plochy mùže skript použít funkcí *OpenWorkspace*. Tato funkce vytváøí plochu právì otevøeného projektu, pokud skript potøebuje vytvoøit plochu jiného projektu, mìl by použít funkci *CreateWorkspaceEx*.

Návratový výraz:
Funkce vrací *WorkspaceID* vytvoøené plochy. Pokud není otevøen žádný projekt, vrací funkce *WorkspaceID -1*, což je pro všechny další funkce neplatné ID.

Knihovna funkcí - prohlížení

Kategorie: Plochy
Téma: CreateWorkspace

Deklarace:
function CreateWorkspace(ProjectID: LongInt; Name, Description: String): LongInt

Parametry:
ProjectID: LongInt
ID projektu, kterému má být plocha vytvoøena
Name: String
Název nové plochy
Description: String
Popis nové plochy

Popis:
Funkce *CreateWorkspaceEx* vytváøí novou plochu projektu *ProjectID* a vrací její identifikátor *WorkspaceID*. Nová plocha bude bude mít název *Name* a popis *Description.* Skript by nikdy nemìl vytváøet plochu bez jména, tedy pøíkazem "*CreateWorkspaceEx(ProjectID, '', '')*". Plocha není po vytvoøení automaticky otevøena, pro otevøení plochy mùže skript použít kombinaci funkcí *OpenProject* a *OpenWorkspace*.

Návratový výraz:
Funkce vrací *WorkspaceID* vytvoøené plochy. Pokud není *ProjectID* platný, vrací funkce *WorkspaceID -1*, což je pro všechny další funkce neplatné ID.

Poznámky:
Pokud potøebuje skript vytvoøit plochu právì otevøeného projektu, nemìl by používat zápis *"CreateWorkspaceEx(GetCurrentWorkspace, Name, Desc)".* Místo toho by mìl k vyuvoøení plochy využít funkci *CreateWorkspace*.